

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SECONDA FACOLTÀ DI INGEGNERIA CON SEDE A CESENA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA BIOMEDICA

TITOLO DELLA TESI

**RICOSTRUZIONE DI IMMAGINI DI
BRONCOSFERE IN MICROSCOPIA
OTTICA CON TECNICHE DI ESTENSIONE
DELLA PROFONDITÀ DI FUOCO**

TESI IN

BIOIMMAGINI E VISIONE LM

Relatore

Prof. Alessandro Bevilacqua

Presentata da

Davide Pollini

Co-Relatori

**Ing. Filippo Piccinini
Dott.ssa Anna Tesei**

Sessione III

Anno Accademico 2010-2011

*Le savant doit ordonner ; on fait la science avec des faits comme
une maison avec des pierres ; mais une accumulation de faits
n'est pas plus une science qu'un tas de pierres n'est une maison.*

*La scienza è fatti; proprio come le case sono fatte di pietre, così
la scienza è fatta di fatti: ma un mucchio di pietre non è una
casa e una collezione di fatti non è necessariamente scienza.*

(Jules Henri Poincaré)

PAROLE CHIAVE

IMAGE FUSION

EXTENDED DEPTH OF FOCUS

NO-REFERENCE METRIC

3-D RECONSTRUCTION

BRONCOSPHERES

Indice

1	Introduzione	1
2	Analisi delle tecniche esistenti	5
2.1	Stato dell'arte	5
2.2	Categorizzazione generale degli algoritmi	7
2.3	Algoritmi di fusione delle immagini	9
2.4	Metodi nel dominio spaziale	10
2.4.1	2.5-D Model-Based Deconvolution	10
2.4.2	Stack Focuser	14
2.4.3	Sharpness Index	18
2.5	Metodi nel dominio frequenziale	21
2.5.1	Complex Wavelet Transform	21
2.5.2	Dual-Tree Complex Wavelet Transform	24
2.5.3	Curvelet Transform	27
3	Nuovo metodo per l'estensione della profondità di fuoco	31
3.1	Schema a blocchi e descrizione	31
3.2	Conversione in scala di grigi attraverso la PCA e formazione dello <i>stack</i> di immagini	34
3.3	<i>Edge detection</i>	35
3.4	Mappa degli indici e filtro di maggioranza	36
3.5	Creazione dell'immagine finale	36
4	Metriche di valutazione	39
4.1	Metriche standard in <i>image processing</i>	39
4.2	Estensione dell'UQI per <i>image fusion</i>	40
5	Materiali	43
5.1	Strumentazione di base di un laboratorio di colture cellulari	44
5.2	Broncosfere	50

5.3	<i>Data set</i> artificiali	52
5.4	<i>Data set</i> reali	53
6	Risultati Sperimentali	57
6.1	<i>Testbed</i> dei parametri dell'algoritmo	57
6.2	Comparazione metodi utilizzando immagini sintetiche	58
6.3	Validazione delle metriche estensione dell'UQI	60
6.4	Nuovo approccio di valutazione	62
6.5	Comparazione metodi utilizzando immagini reali	65
6.6	Metodo di ricostruzione 3-D	72
7	Interfaccia grafica	75
7.1	Strumenti MATLAB per lo sviluppo	75
7.2	Progetto e realizzazione	76
8	Conclusioni e sviluppi futuri	85
A	Trasformata Wavelet e Trasformata Curvelet	89
B	Codici MATLAB	99
B.1	Algoritmo	99
B.2	Parametri	103
	Bibliografia	105
	Ringraziamenti	111

Capitolo 1

Introduzione

Gli oggetti osservati al microscopio ottico risultano essere spesso più spessi della *profondità di fuoco* (conosciuta anche con il nome di *profondità di campo*) del sistema e questo significa che per questi oggetti che non è possibile acquisire una singola immagine completamente a fuoco. In particolare, questo problema risulta essere ben conosciuto in biologia, dove ad esempio utilizzando microscopi comuni risulta impossibile acquisire immagini relative a sferoidi multicellulari senza problemi di regioni non a fuoco.

Per risolvere questo problema fin dagli anni '70 sono stati proposti innumerevoli metodi per estendere la profondità di fuoco dei microscopi attraverso tecniche di fusione di immagini. In generale, con il termine fusione di immagini (dall'inglese *image fusion*) vengono indicati quei processi atti a combinare le diverse informazioni provenienti da due o più immagini di una stessa scena, acquisite con lo stesso o eventualmente anche diversi tipi di sensore, al fine di formare un'immagine composta finale che tipicamente risulta essere più informativa, visivamente migliore, meno rumorosa e più adatta per essere utilizzata per eventuali successive elaborazioni. In particolare, l'obiettivo principale dei processi di fusione di immagini risulta essere ridurre rumore e minimizzare la ridondanza di informazioni nell'immagine finale, ma allo stesso tempo massimizzare le informazioni particolarmente rilevanti per un eventuale specifica un'applicazione o *task*. In questo lavoro di Tesi, in particolare, è stato trattato e analizzato un singolo sottocampo dell'*image fusion*, quello riguardante la fusione di immagini ottenute utilizzando uno stesso sensore ma acquisendo immagini della stessa scena a distanze focali differenti.

Tipicamente, tutti i metodi proposti in letteratura per estendere al

profondità di fuoco dei microscopi, sfruttano uno *stack* di immagini, acquisite manualmente semplicemente muovendo il piano porta oggetto del microscopio nella direzione z dell'asse ottico in modo da ottenere sezioni ottiche chiamate *slice*. Se le dimensioni degli oggetti sono molto maggiori della profondità di fuoco del sistema, solo una piccola porzione delle immagini acquisite risulterà a fuoco. In generale, gli algoritmi proposti per estendere la profondità di fuoco cercheranno di rilevare le regioni a fuoco nelle singole immagini e di fondere le informazioni al fine di ottenere una singola immagine finale completamente a fuoco.

A causa del vasto numero di metodi proposti e la mancanza di un approccio validato per compararli, risulta molto difficile, specialmente per biologi e medici senza conoscenze di elaborazione di immagini, capire quale metodo potrebbe fornire migliori prestazioni per una specifica applicazione.

In questo lavoro di tesi è stato prima proposto e validato un approccio di valutazione estensiva capace di comparare e classificare i vari metodi presentati in letteratura per estendere la profondità di campo di un microscopio. Inoltre, è stato studiato un nuovo approccio basato su un'analisi spaziale dei valori di derivata dei singoli *pixel* in modo da rilevare le regioni a fuoco nelle immagini dello *stack* di partenza sfruttate poi per comporre l'immagine finale completamente a fuoco. Gli esperimenti condotti sfruttando *stack* di immagini reali e sintetiche provano che l'algoritmo proposto è in grado di superare le *performance* ottenibili con quelli allo stato dell'arte.

In particolare, questo lavoro di Tesi si inserisce nel contesto di una consolidata collaborazione in atto tra il Laboratorio di Bioscienze dell'Istituto scientifico Romagnolo per lo Studio e la cura dei Tumori (IRST) e il gruppo di ricerca *Computer Vision Group* (CVG) dell'Università di Bologna. L'obiettivo principale del progetto, STAMINAL (caratterizzazione di cellule STAminali tramite supporto per l'analisi autoMatIca delle immaginiNi microscopiche in terApia pre-cLinica) è la caratterizzazione di aggregati multicellulari tramite l'impiego di tecniche di *image processing* applicate ad immagini acquisite con microscopi ottici tradizionali. In particolare si vuole studiare la morfologia e la funzionalità di aggregati multicellulari a forma sferica (sferoidi) di cellule provenienti dai bronchi alveolari (broncosfere) a seguito di vari trattamenti biologici e radioterapici. Una ricerca guidata dalla Dott.ssa Anna Tesei, afferente al Laboratorio di Bioscienze dell'IRST, coordinato dal Dott. Wainer Zoli, è stata la prima al mondo ad essere

riuscita ad isolare le broncosfere [44]. L'essere riusciti ad isolare e ricreare degli sferoidi *in vitro* è un risultato molto importante in quanto risultano essere un ottimo modello allo stato solido di tessuto umano. In particolare, gli sferoidi ottenuti da linee stabilizzate di cellule tumorali risultano essere un ottimo modello di tumore su cui testare farmaci e sperimentare terapie di cura, quali ad esempio i parametri di esposizione in radioterapia. Infatti, gli sferoidi di culture tumorali trattati sono stati utilizzati anche in radioterapia sperimentale per valutare l'effetto di regimi di irraggiamento singoli e/o frazionati da soli o in combinazione con farmaci e/o radiosensibilizzanti. Mentre attualmente in letteratura esistono molti articoli sulla caratterizzazione di sferoidi tumorali multicellulari usati in studi di terapia combinatoria orientata [13], [15], [27], [34], [51], nessuno finora ha cercato di identificare la loro struttura interna ed esterna attraverso studi densitometrici e la ricostruzione 3-D.

Il metodo proposto per l'estensione della profondità di campo ha risposto in maniera positiva alle esigenze espresse dai biologi di IRST di riuscire ad acquisire immagini degli sferoidi evitando la presenza di regioni non a fuoco e di visualizzarne la superficie 3-D. In particolare, per favorire la visualizzazione 3-D è stata creata un'interfaccia grafica estremamente *user's friendly* che ha soddisfatto a pieno le richieste.

Questa elaborato di Tesi è composto dai seguenti capitoli:

- Capitolo 2: presenta una breve descrizione dello stato dell'arte relativo alle tecniche di *image fusion* utilizzate in microscopia per estendere la profondità di fuoco.
- Capitolo 3: descrive nei dettagli il nuovo approccio proposto per l'estensione della profondità di campo.
- Capitolo 4: riporta le principali metriche *standard* di valutazione utilizzate comunemente in elaborazione di immagini e la loro specifica estensione per il campo dell'*image fusion*.
- Capitolo 5: descrive il *set-up* degli strumenti utilizzati per creare e/o acquisire le sequenze di immagini parzialmente a fuoco utilizzate per valutare e confrontare la qualità dei risultati dei diversi algoritmi.
- Capitolo 6: descrive il *test-bed* dell'esperimento e dei parametri dell'algoritmo proposto discutendo e analizzando in dettaglio i risultati sperimentali

ottenuti con le diverse sequenze di immagini e le diverse metriche di valutazione utilizzate.

- Capitolo 7: presenta l'interfaccia grafica progettata e sviluppata al fine di fornire ai biologi uno strumento estremamente *user's friendly* con cui ottenere immagini completamente a fuoco.

Capitolo 2

Analisi delle tecniche esistenti

Fin dagli anni '70 sono stati proposti metodi per la ricostruzione di immagini 2D sfruttando informazioni sulla profondità di campo, chiamati in inglese metodi *extended depth of focus* (o *depth of field*, DOF). Gli approcci proposti in letteratura sono molti ed estremamente vari, sia per prestazioni che per costi computazionali e dominio di lavoro. In questo capitolo prima sarà proposta una classificazione basata sul dominio di lavoro dell'algoritmo, tipicamente spaziale o frequenziale. Poi saranno descritti in dettaglio alcuni metodi implementati e gratuitamente distribuiti in ImageJ (<http://rsbweb.nih.gov/ij/>), *software open source* ampiamente utilizzato per l'elaborazione delle immagini. Infine, saranno presentati alcuni approcci di ultima generazione basati su un'analisi delle immagini condotta nel dominio delle frequenze, sfruttando wavelet e curvelet che lavorano attraverso un'analisi multi risoluzione del segnale.

2.1 Stato dell'arte

La *image fusion* (fusione delle immagini) è una sottobranca della più generale *data fusion* (fusione di dati), in cui gli elementi di base sono costituiti da dati rappresentati in forma di *array* che rappresentano luminosità, colore, temperatura, distanza e altre proprietà della scena rappresentata. Tali dati possono essere bidimensionali (immagini), tridimensionali (immagini volumetriche), o eventualmente anche di dimensioni superiori (*stack* di immagini multicanale). I primi lavori sulla fusione delle immagini risalgono alla metà degli anni ottanta. Il primo metodo utilizzato per fondere più immagini relative alla stessa scena era basato sulla semplice media calcolata *pixel per pixel* tra le varie immagini sorgenti. Questo metodo, banale da

implementare e veloce da eseguire, sfortunatamente tende a sopprimere le caratteristiche salienti che dovrebbero essere mantenute nell'immagine finale, producendo come risultato immagine caratterizzate da un indesiderato basso contrasto. In seguito, Pieper e Korpel [38], proposero un approccio basato su un operatore a differenza non direzionale per identificare le aree a fuoco nelle singole immagini originali: prima tutte le *slide* sono "derivate" utilizzando l'operatore sopracitato, poi per ogni singolo *pixel* viene identificata nello *stack* l'immagine caratterizzata dal più alto valore di "derivata". L'immagine finale viene infine costruita copiando dalle varie immagini iniziali i valori di *pixel* identificati scegliendo i valori massimi identificati dall'operatore a differenza. Nello stesso periodo, Sugimoto e Ichioka [43], proponevano un metodo basato sulla media pesata delle immagini, in cui i pesi erano calcolati come rapporto tra la varianza dell'immagine (definita come sommatoria delle varianze locali calcolate in una finestra $n \times m$) e la varianza totale cioè la sommatoria di tutte le varianze delle immagini. I primi metodi definiti *area based* o *neighborhood based*, dove per la selezione della *slice* a fuoco in quella determinata posizione $x-y$ veniva valutata una funzione calcolata non sul singolo *pixel*, ma su una finestra di dimensioni $n \times n$, furono introdotti da Tympel [46] e Goldsmith [20]. Le funzioni utilizzate dai due metodi sono molto simili tra di loro. In particolare, entrambe si basano sulla somma delle differenze tra il *pixel* centrale e gli altri *pixel* all'interno della finestra.

Successivamente, Burt [9],[4] fu uno dei primi a riportare l'uso di tecniche di decomposizione basate sull'utilizzo della piramide laplaciana (in inglese *Laplacian Pyramid*, LP) per la fusione delle immagini. La LP è chiamata così perchè in essa la decomposizione delle immagini è strutturata in maniera piramidale: ogni livello successivo è sottocampionato (in inglese *subsampling*) di un fattore 2 e, per ottenere le bande ad alta frequenza si sottrae all'immagine originale una sua copia sfuocata con un filtro gaussiano. Allo stesso tempo, Adelson [12] approfondì metodi per l'uso della tecnica del laplaciano nella costruzione di un'immagine finale con estensione della profondità di campo da un set di immagini acquisite con una macchina fotografica fissa ma con differenti lunghezze focali. L'uso della trasformata wavelet discreta (in inglese *Discrete Wavelet Transform*, DWT) per la fusione d'immagini fu quasi simultaneamente proposta da Li *et al.* [29] e Chipman *et al.* [4], [25]. Nel frattempo Koren *et al.* [4], [50] descrissero la trasformata wavelet diadica e orientabile (in inglese *Steerable Dyadic Wavelet Transform*) per la fusione d'immagini. Contemporaneamente Waxman e colleghi [4], [47] svilupparono una metodologia per la fusione d'immagini basata su modelli

biologici della visione del colore.

Infine, negli anni più recenti, l'attenzione si è spostata sull'utilizzo della trasformata wavelet complessa (in inglese *Complex Wavelet Transform*, CWT) [17] e della trasformata wavelet complessa a doppio albero (in inglese *Dual Tree Complex Wavelet Transform*, DT-CWT) [42], [24] che hanno permesso di superare i maggiori problemi legati alla DWT come la varianza alle traslazioni, l'*aliasing* e la mancanza selettiva di direzionalità. La tecnica che al giorno d'oggi sembra poter fornire risultati migliori, anche rispetto alla CWT e DT-CWT, è la trasformata curvelet (in inglese *Curvelet Transform*) [10],[32] che grazie alla sua maggiore direzionalità permette di analizzare e rappresentare i contorni in modo molto efficiente.

2.2 Categorizzazione generale degli algoritmi

Gli algoritmi per la fusione d'immagini DOF sono tipicamente categorizzati in base al dominio di lavoro, tipicamente spaziale o frequenziale. Lavorare nel dominio spaziale di un'immagine significa agire direttamente sui valori dei *pixel* che la costituiscono. Invece, operare nel dominio delle frequenze significa applicare una specifica trasformata all'immagine, abbandonando così il classico dominio spaziale dove tutti i punti sono georeferenziati con coordinate scalari. Nel dominio delle trasformate l'immagine non è più rappresentata con valori in scala di grigi o dai vettori RGB per ogni singolo *pixel*, ma a ogni punto è associato un valore scalare, definito come coefficiente della trasformata. Se nel dominio delle trasformate, durante il processo di fusione, viene modificato anche uno solo di questi coefficienti, quando si esegue l'antitrasformata per riportare l'immagine nel dominio spaziale, può accadere che i valori di una parte o dell'intera immagine risultino fortemente modificati. Quindi, agendo sui coefficienti nel tentativo di migliorare l'immagine in alcune zone, si potrebbero creare artefatti non desiderati in altre parti di essa. Gli algoritmi che lavorano nel dominio spaziale hanno l'abilità di concentrarsi su determinate aree dell'immagine limitando i cambiamenti nelle aree circostanti: non vanno quindi a interferire con i valori dei *pixel* che circondano la regione dell'immagine processata.

Entrando nel dettaglio dei metodi che lavorano nel dominio delle frequenze, studi scientifici ([17], [46]) hanno dimostrato che le metodiche hanno portato ai risultati migliori sono quelli basati sulla decomposizione multiscala. La decomposizione multiscala, o analisi multi risoluzione, è una

tecnica di analisi dei segnali basata su una trasformata multilivello (dalla quale ha origine il termine “multiscala”) in cui il segnale viene scomposto a diversi livelli di scala. In altre parole, il segnale è analizzato a diversi livelli di dettaglio, in particolare si cerca di separare il contenuto a bassa frequenza, che rappresenta la struttura approssimativa dell’architettura del segnale, dai contenuti ad alta frequenza rappresentanti i dettagli dello stesso. I metodi basati sulle trasformate multiscala sono risultati utili per l’analisi delle informazioni contenute nelle immagini, ed in particolare sono stati sfruttati per la fusione delle immagini stesse. L’idea alla base di questi approcci per fusione, è eseguire la trasformata multiscala per ogni singola immagine sorgente, tramite delle regole estrarre da ogni singola immagine le aree a fuoco e costruire con esse un’unica rappresentazione multiscala che costituisce l’immagine composita finale nel dominio delle trasformate. Infine, l’immagine “fusa” nel classico dominio spaziale è ottenuta attraverso un’operazione di antitrasformazione, cioè è eseguita la trasformata multiscala inversa.

Nel gruppo dei metodi che lavorano nel dominio spaziale possono essere classificati una grande varietà di approcci basati su differenti tecniche di analisi delle immagini. Vista la vastità delle differenze tra i vari metodi proposti in letteratura risulta difficile riassumere una idea di base. Comunque nelle prossime sezioni saranno descritti nei particolari alcuni esempi di metodi considerati rappresentativi.

Per concludere, possiamo dire che tipicamente l’idea comune alla base dei vari approcci per fusione di immagini, sia quelli che lavorano nel dominio spaziale, sia quelli che lavorano nel dominio delle frequenze, è il valutare quali regioni delle immagini di partenza possono fornire informazioni utili alla costruzione dell’immagine composita finale, cercando di quantizzare la qualità delle varie singole regioni di ogni immagine, attraverso indici metrici generalmente chiamati *misura del livello di qualità*, ed utilizzare nella ricostruzione finale solo le regioni classificate ad alto contenuto informativo. Generalmente, con il termine *fusione delle immagini* (in inglese *image fusion*) s’intende la combinazione del contenuto d’immagini riguardanti la stessa scena e provenienti o da diversi sensori o dallo stesso sensore, in modo che l’immagine finale abbia un contenuto informativo nuovo o comunque superiore a quello che ciascuna delle immagini singolarmente può dare. In questo lavoro di tesi si è voluto confrontare metodi di fusione delle immagini disegnati per sfruttare immagini prese da uno stesso sensore a piani focali differenti.

2.3 Algoritmi di fusione delle immagini

In totale sono stati analizzati sette differenti algoritmi, di cui quattro sviluppati in linguaggio java e rilasciati come *plug-in* gratuiti per ImageJ (<http://rsbweb.nih.gov/ij/>), *software open source* di elaborazione d'immagini, mentre i restanti sono stati sviluppati in codice MATLAB (© The MathWorks, Inc., Massachusetts, USA). La scelta di analizzare, oltre ai *software* creati in MATLAB ©, i metodi implementati in ImageJ, è stata dettata dal fatto che questo *software* per l'elaborazione d'immagini digitali è: ampiamente utilizzato in letteratura, *open source*, di facile reperibilità, progettato con una *open architecture* che prevede la possibilità di avere estensioni tramite *plug-in* e che nonostante sia un *software user's friendly* e molto vasto e completo. Tutti gli algoritmi che sono stati analizzati, necessitano che le singole immagini dei diversi piani focali siano impilate in un unico file che è definito *stack*. Lo *stack* non è altro quindi che un *array* espandibile d'immagini che devono necessariamente essere della stessa dimensione, allineate fra loro e gli oggetti fotografati non devono essere in movimento. Generalmente ci si riferisce alle immagini che formano uno *stack* con il termine *slice*.

Seguendo la classificazione nei due grandi gruppi precedentemente descritti, i 7 algoritmi utilizzati possono essere così suddivisi:

- Metodi di fusione che lavorano nel dominio spaziale:
 1. *2.5-D Model-Based Deconvolution* (MDB), François Aguet, sviluppato come *plug-in* ImageJ [2],
 2. *Stack Focuser* (SF), prof. Mikhail Umorin, sviluppato come *plug-in* di ImageJ,
 3. *Sharpness Index* (SI), Emerson Ferreira de Lucena, Luis Rogerio de Oliveira Hein, sviluppato come *plug-in* di ImageJ, ispirandosi all'articolo di Goldsmith [20],
 4. *Analisi Puntuale Locale* (AnPuLo), sviluppato interamente da me all'interno del lavoro di tesi (vedi Capitolo 3).

- Metodi di fusione che lavorano nel dominio frequenziale:
 1. *Complex Wavelet Transform* (CWT), Alex Prudencio, Jesse Berent and Daniel Sage, sviluppato come *plug-in* ImageJ [17],

2. *Dual-Tree Complex Wavelet Transform* (DT-CWT) implementato da me in codice MATLAB ©, sfruttando funzioni reperibili gratuitamente *online*,
3. *Curvelet Transform* (CT) implementato da me in codice MATLAB ©, sfruttando funzioni reperibili gratuitamente *online*.

2.4 Metodi nel dominio spaziale

2.4.1 2.5-D Model-Based Deconvolution

Il metodo MDB, a differenza degli altri metodi che combinano le informazioni a fuoco provenienti da diverse immagini dello *stack*, per formare un'unica immagine che raffiguri l'intero campione, stima in maniera congiunta la *texture* e la topografia del campione, basandosi sul modello di formazione dell'immagine descritto attraverso la convoluzione di un modello matematico del campione e la risposta all'impulso, detta *point spread function* (PSF). I microscopi ottici a campo chiaro, a seconda delle proprietà ottiche del campione vengono configurati per lavorare o con luce trasmessa o con luce riflessa (definite rispettivamente configurazione *diascopica* e *episcopica*). La modalità a trasmissione è appropriata per i campioni che sono sufficientemente trasparenti (ad esempio provini istologici di tessuto umano) mentre, la modalità a riflessione consente di visualizzare campioni opachi (ad esempio provini metallici). In particolare, nell'algoritmo è stato proposto un modello adatto a processare immagini acquisite con entrambe le modalità. Per un campione opaco visualizzato in riflessione, l'oggetto può essere modellato come una superficie 3-D dato che la luce non riesce a penetrare a causa della sua opacità. Si è scoperto che questo modello di superficie sottile è una valida approssimazione anche per la configurazione diascopica, a condizione che: il campione sia sufficientemente spesso, in modo tale che solo la sua superficie ricada all'interno della profondità di campo del microscopio, e quindi appaia a fuoco; e si ignorino i contributi di tutte le zone che, seppur colpite dalla luce dato che il campione è trasparente e il microscopio è usato in modalità a trasmissione, sono oltre la profondità di fuoco del sistema. Per poter essere applicata questa approssimazione, il campione trasparente deve essere significativamente più spesso rispetto alla profondità di fuoco del sistema che è definita come:

$$d = \frac{\lambda n_i}{NA^2} \quad (2.1)$$

Dove λ è la lunghezza d'onda dell'illuminazione, n_i è l'indice di rifrazione del mezzo, e NA indica l'apertura numerica dell'obiettivo.

Nel metodo MBD la ricostruzione dell'immagine è formulata come un problema ai minimi quadrati, e la soluzione è ottenuta tramite l'ottimizzazione iterativa di *texture* e topografia. In questo modo la topografia non è più limitata a valori discreti, ma può assumere valori continui fino a combaciare perfettamente con le misure.

Il campione $o(x, y, z)$ può essere espresso come una superficie 3-D descritta da una topografia $p(x, y)$ sulla quale è mappata una *texture* $f(x, y)$:

$$o(x, y, z) = f(x, y)\delta(z - p(x, y)) \quad (2.2)$$

Dove la distribuzione di Dirac rappresenta la superficie. La formazione dell'immagine e il processo di acquisizione sono modellati come una convoluzione tra l'oggetto e la PSF 3-D del microscopio $h(x, y, z)$, ottenendo il volume di acquisizione $\tilde{s}(x, y, z)$:

$$\tilde{s}(x, y, z) = \int_{R^2} f(u, v)h(x - u, y - v, z - p(u, v))dudv \quad (2.3)$$

Nella pratica quest'operazione è seguita da un campionamento in modo da creare i dati discreti (Figura 2.1). La funzione costo quadratica utilizzata

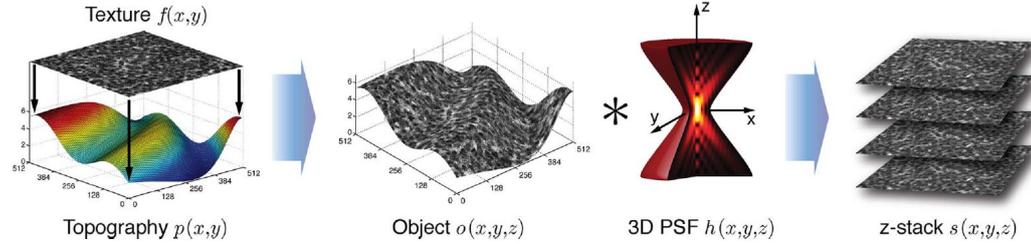


Figura 2.1: Illustrazione della formazione dell'immagine. Cortesemente da [2]

per ottenere la topografia e la *texture* è la seguente:

$$J(f, p) = \int_{R^3} (s(x, y, z) - \tilde{s}(x, y, z))^2 dx dy dz \quad (2.4)$$

Attraverso questa formulazione si può vedere che la stima della topografia dipenderà dalla *texture*, e viceversa. Perciò per la risoluzione si è dovuti ricorrere a un metodo iterativo di ottimizzazione in due fasi che alternativamente aggiorna la *texture* e la topografia, dove la *texture* e la topografia all'iterazione τ -esima sono indicate rispettivamente come $f^{(\tau)}$ e $p^{(\tau)}$. Il mo-

dello prevede di partire da una stima iniziale della topografia $p^{(\tau)}$ per stimare la *texture* minimizzando la funzione costo della Equazione (2.4) rispetto ad f :

$$f^{(\tau+1)} = \arg \min_f J(f, p^{(\tau)}) \quad (2.5)$$

Il minimo non può essere trovato in forma chiusa, per cui per arrivare a una soluzione si è applicato il metodo dei gradienti su J rispetto a $f(x, y)$. Il modello esatto però non è favorevole dal punto di vista computazionale, per cui l'algoritmo è stato implementato utilizzando un modello semplificato e separabile. Si consideri il modello più semplice di PSF che soddisfi il modello di formazione dell'immagine semplificato ipotizzato in precedenza. Il modello

$$h(x, y, z) = \delta(x)\delta(y)h(z) \quad (2.6)$$

Può essere messo in relazione all'ipotesi che le informazioni a fuoco sono correlate con l'intensità lungo l'asse z . Applicando questa semplificazione la funzione costa globale può essere decomposta e quindi la minimizzazione può essere eseguita in modo disaccoppiato attorno al punto (x, y) . In modo simile anche la topografia può essere aggiornata. A tal fine, si minimizza la funzione costo rispetto alla topografia utilizzando l'ultima stima della *texture* $f^{(\tau+1)}$:

$$p^{(\tau+1)} = \arg \min_p J(f^{(\tau+1)}, p) \quad (2.7)$$

Purtroppo anche in questo caso la risoluzione del modello esatto non è ottimale a causa della diretta dipendenza della topografia dalla stima corrente della *texture* $f^{(\tau+1)}$ che può perturbare l'algoritmo nella sua fase iniziale quando sono messi in atto forti aggiustamenti globali che posso far convergere verso un errato ottimo locale. Per affrontare questa potenziale instabilità, si esegue la stima della topografia utilizzando una struttura da grezzo-a-fine (in inglese *coarse-to-fine framework*). Si rappresenta la topografia a una risoluzione m in basi invarianti alla traslazione generate attraverso B-spline simmetriche di ordine n dilatate da $m \in N^+$:

$$p_m = \sum_{k \in Z^2} c_m[k] \beta^n \left(\frac{X}{m} - k \right) \quad (2.8)$$

Con $X = (x, y)$. Il problema di minimizzazione della Equazione (2.7) allora equivale a trovare i coefficienti $c_m[k]$ ottimali per la B-spline. L'algoritmo per la stima congiunta della *texture* e della topografia parte da un'inizializzazione della *texture*, e itera tra la stima della topografia e la rifinitura della *texture*.

Quando è raggiunta la convergenza a quella scala, l'algoritmo passa alla scala più fine ($m' = m/2$ nel caso diadico) e replica il processo. Il tutto termina quando la convergenza è raggiunta per la scala $m = 1$ (Figura 2.2) [2]. L'interfaccia grafica (vedi Figura 2.3) è suddivisa in quattro aree: l'area

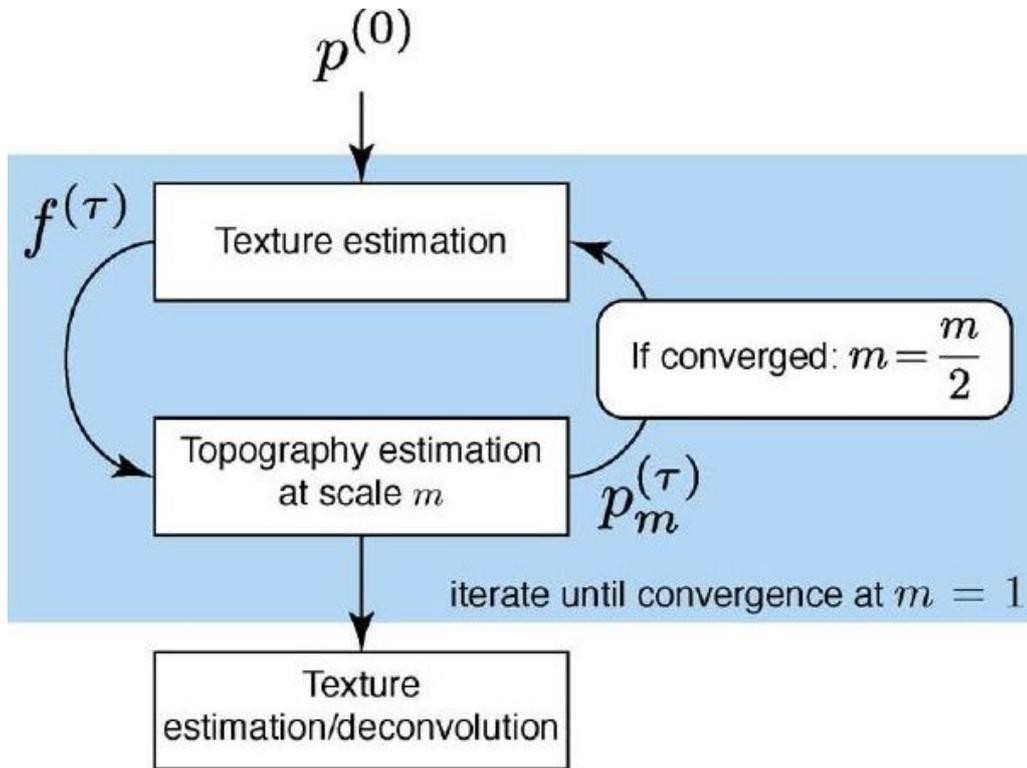


Figura 2.2: Rappresentazione schematica dell'algoritmo di ottimizzazione. Cortesemente da [2].

per la modifica dei parametri della PSF, l'area per la creazione dei dati artificiali, l'area per la ricostruzione della *texture* e della topografia e l'area per la visualizzazione 3-D. Per ottenere l'immagine completamente a fuoco bisogna impostare i parametri della PSF (Figura 2.3-1) e poi utilizzare l'area per il calcolo della *texture* e della topografia (Figura 2.3-3). In realtà, gli autori, fanno notare che essendo il problema dell'ottimizzazione della *texture* e della topografia, un problema ben posto, si può supporre che l'algoritmo di ricostruzione possa essere relativamente insensibile alla parametrizzazione della PSF. In realtà, solo la PSF $h(x, y, 0)$ è necessario che sia precisa. In caso contrario, a causa dell'ipotesi fatta sul campione, la deconvoluzione per la creazione dell'immagine fusa è o troppo debole, o produce informazioni fuori fuoco che sono falsamente riassegnate alla *texture*. L'approssimazione gaussiana della PSF si comporta molto bene nelle regioni

a fuoco e, conseguentemente, non influisce significativamente sui risultati. Questo fa sì che ci si possa quasi disinteressare dei parametri η_0 e η_1 che determinano la varianza della PSF perché è stato provato dagli autori che un errore fino al 30% su η_1 permette lo stesso una ricostruzione senza una perdita significativa di accuratezza [2]. Infatti, nella nostra analisi abbiamo mantenuto i valori di *default* ottenendo comunque ottimi risultati.

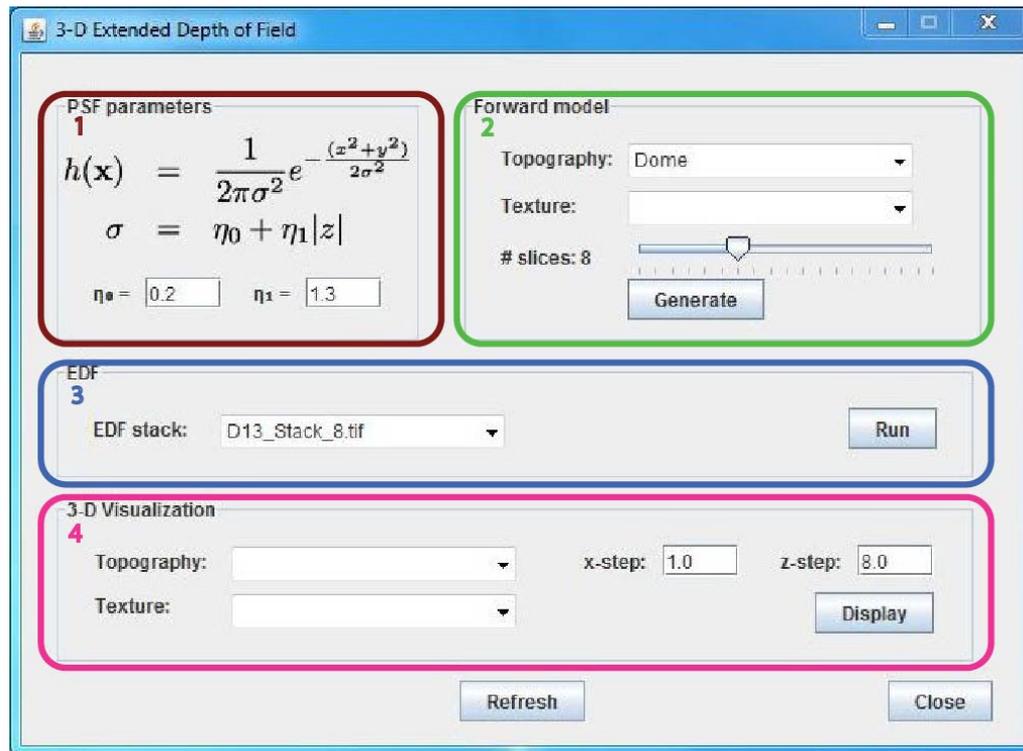


Figura 2.3: Interfaccia grafica del metodo 2.5-D-MBD. (1) Area per la modifica dei parametri della PSF; (2) area per la creazione di stack artificiali; (3) area per la ricostruzione della texture e della topografia; (4) area per la visualizzazione in 3-D.

2.4.2 Stack Focuser

Il metodo SF, sviluppato dal Prof. Mikhail Umorin di Anatomia e Fisiologia al Brookhaven College (Farmers Branch, Texas), fu rilasciato nella sua prima versione nell'ottobre del 2002 e in seguito è stato migliorato fino all'attuale ultima versione risalente al luglio del 2006. Il *software* è gratuitamente scaricabile dalla sezione *Plug-in* di ImageJ (<http://rsbweb.nih.gov/ij/plugins/>). Nel sito, oltre al file compilato pronto per l'installazione, è presente anche il codice sorgente che permette di studiare nel dettaglio il

funzionamento dell'algoritmo. Il metodo SF è in grado di lavorare su *stack* d'immagini a 8-bit, 16-bit e a colori. Se le immagini da analizzare sono già in scala di grigi il primo passo dell'algoritmo consiste nel creare una copia a 32-bit in virgola mobile dello *stack* originale in modo da preservare la precisione; se invece le immagini sono a colori, prima si scompone lo *stack* nei tre canali (rosso, verde e blue) e poi si applica per ogni canale il passo precedentemente descritto. Solo alla fine di tutto il processo di fusione, se le immagini originali erano a colori, i tre singoli canali sono riuniti per ottenere l'immagine finale a colori. In uno degli ultimi aggiornamenti è stata aggiunta l'opzione di usare la messa a fuoco solo su un colore, il verde, e di copiare il risultato anche per gli altri colori. Questo permette di velocizzare il programma. Il corpo centrale dell'algoritmo, qualsiasi sia la tipologia delle immagini in ingresso, è quindi eseguito su immagini in scala di grigi.

Dopo aver creato la copia dello *stack* originale, il metodo prosegue filtrando ogni singola *slice* con un filtro mediano 3×3 , con un filtro a media 3×3 e poi utilizzando Sobel per il rilevamento dei contorni (*edges*). L'operatore di Sobel applica due *kernel* 3×3 , cioè due matrici di convoluzione, per calcolare valori approssimati delle derivate orizzontali e verticali. Se chiamiamo I l'immagine sorgente, e G_x e G_y le due immagini i cui valori rappresentano rispettivamente la valutazione approssimata delle derivate in orizzontale ed in verticale, l'operazione è descritta da:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (2.9)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.10)$$

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.11)$$

Dopo l'operatore di Sobel, ogni singola *slice* così pre-processata, è fornita in ingresso a una funzione chiamata *maxFilter* che, oltre all'immagine sorgente in cui sono stati estratti i contorni, ha altri due ingressi: una matrice delle stesse dimensioni dell'immagine a cui è associata un mappa dei colori, nel caso specifico è utilizzata la *look-up table* (LUT) predefinita della scala di grigi, e la dimensione del *kernel*. La funzione *maxFilter* analizza *pixel* per *pixel* l'immagine sorgente e per ogni singolo *pixel* richiama un'altra funzione

chiamata *findMaxInNeigh* che prende in ingresso l'immagine sorgente, le coordinate x e y del *pixel* che si sta analizzando e la dimensione del *kernel*. Questa funzione crea una regione d'interesse (in inglese *region of interest*, ROI) quadrata di lato pari alla dimensione del *kernel* e centrata nel *pixel* di coordinate x e y . È scandagliato ogni punto della ROI e come risultato finale si ottiene il valore massimo della regione e questo valore viene assegnato al *pixel*(x, y). Dopo che questa cascata di funzioni è stata applicata a tutte le *slice* dello *stack* si ha un nuovo *stack*, con lo stesso numero di *slice*, che hanno le stesse dimensioni delle immagini originali, dove ogni *pixel* di ogni singola immagine è il valore massimo di una ROI quadrata $n \times n$ centrata su quel *pixel*. Su questo nuovo *stack* è usata la funzione chiamata *pasteGreyImage* che confronta fra di loro i valori dello stesso *pixel* ma nelle diverse *slice* dello *stack* e cerca in quale *slice* il *pixel* ha ottenuto il valore maggiore e, successivamente crea una matrice in cui ogni elemento rappresenta la coordinata z dell'immagine all'interno dello *stack*. Questa matrice guida la costruzione dell'immagine finale perché ogni suo elemento (x, y) indica da quale *slice* bisogna prendere il *pixel*(x, y) per ottenere tutta la scena completamente a fuoco. L'interfaccia grafica (Figura 2.4) risulta essere molto semplice con pochi parametri e opzioni su cui poter interagire. L'unico parametro che può essere modificato è la dimensione del *kernel*, di *default* è fissata a 11 ma può essere fatta variare tra tutti i numeri dispari maggiori di uno. Delle due opzioni che è possibile attivare, una riguarda la possibilità di ottenere, oltre all'immagine completamente a fuoco, un'altra immagine che viene chiamata mappa delle altezze, in inglese *depth map* (Figura 2.5), che riporta per ogni posizione x - y l'indice della *slice*, espresso in toni di grigio come mostrato nell'Equazione (2.10), dalla quale è stato copiato il valore nel *pixel* dell'immagine composita finale:

$$\text{tono di grigio} = \frac{255 * i}{n. \text{ tot. di slice}} \quad (2.12)$$

Dove i rappresenta la *slice* da cui è stato preso quel determinato *pixel*. L'altra opzione riguarda le immagini a colori e in particolare la possibilità già descritta in precedenza di poter usare per la messa a fuoco solo un canale.

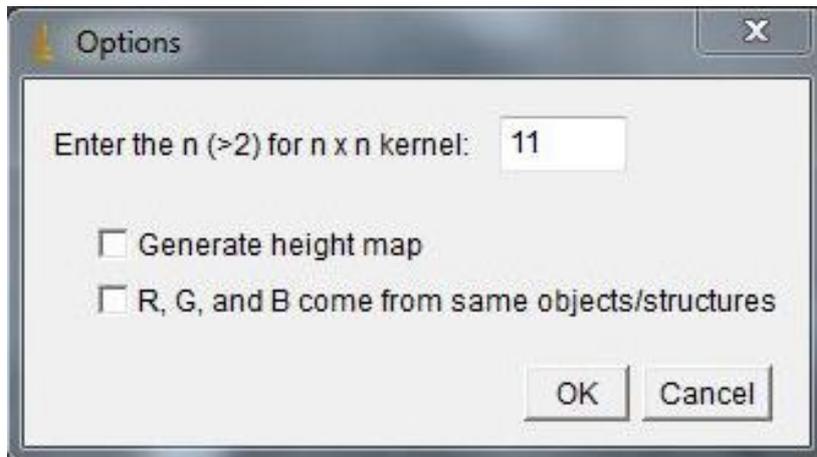


Figura 2.4: Interfaccia grafica di ImageJ del metodo Stack Focuser.

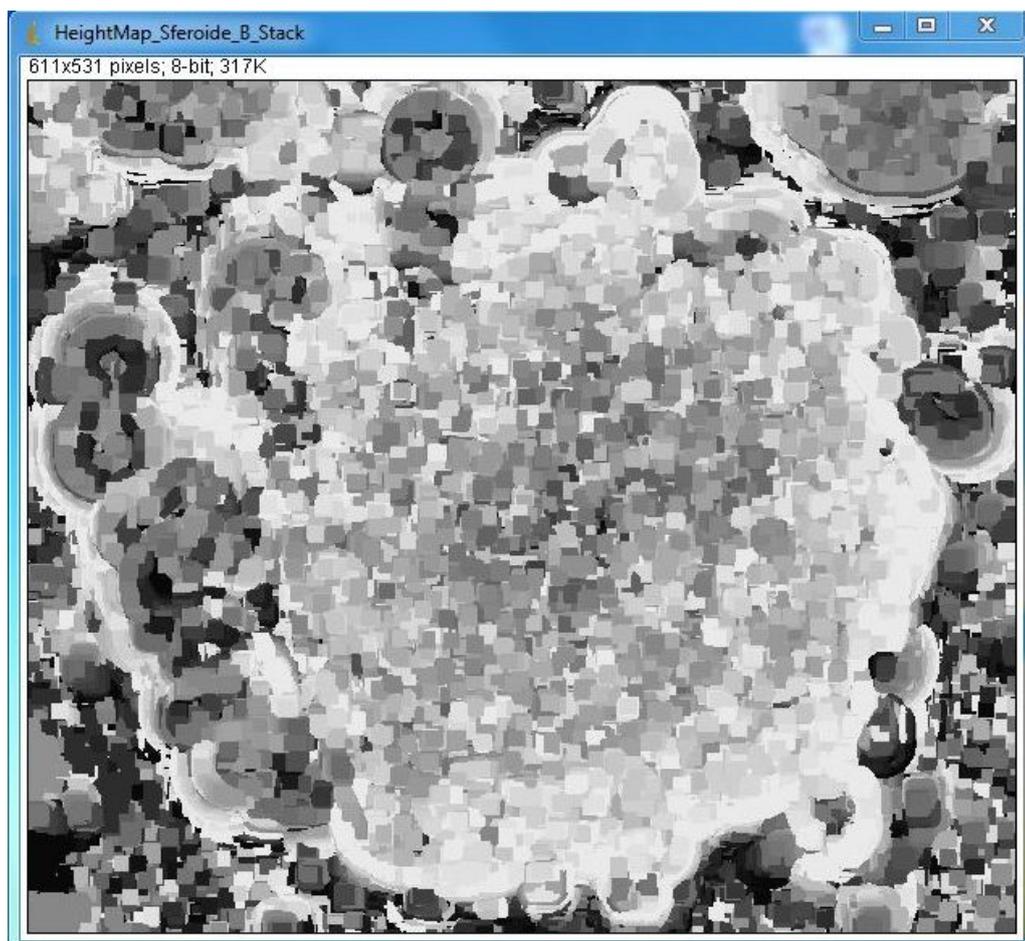


Figura 2.5: Immagine della mappa delle altezze ottenuta attivando l'opzione *Generate height map*.

2.4.3 Sharpness Index

L'algoritmo è stato sviluppato da Dr. Emerson Ferreira de Lucena e dal Prof. Rogerio de Oliveira Luis Hein. Per l'implementazione si sono ispirati all'articolo scritto da Goldsmith nel 2000 [20]. Anche questo metodo deve ricevere in ingresso uno *stack* d'immagini, ma a differenza di quello precedentemente descritto accetta solo immagini in scala di grigi. L'algoritmo proposto da Goldsmith prevede la creazione di un *array* inizializzato a zero e utilizzato per immagazzinare gli indici di nitidezza (*sharpness indexes*) relativi a ogni singolo *pixel* dell'immagine, e due immagini, una per contenere l'immagine finale in cui tutta la scena è a fuoco, iterativamente costruita, chiamata anche immagine composita, e una per l'immagine definita come immagine delle profondità o *depth map*. Il processo procede *pixel* per *pixel* e immagine per immagine, finché lo *stack* di immagini non è stato interamente processato. Ogni volta che i *pixel* di un'immagine forniscono valori degli indici di nitidezza maggiori dei valori immagazzinati nell'*array*, il *software* esegue le seguenti azioni:

1. copia all'interno dell'immagine composita il valore dell'intensità del *pixel* alla posizione corrente nell'immagine sorgente,
2. sostituisce il valore dell'indice di nitidezza immagazzinato nell'*array* con quello appena calcolato se maggiore del precedente,
3. se e solo se il valore è stato sostituito, viene aggiornato l'indice della *slice*, equivalente alla posizione occupata dall'immagine appena processata all'interno dello *stack*, all'interno dell'immagine delle profondità.

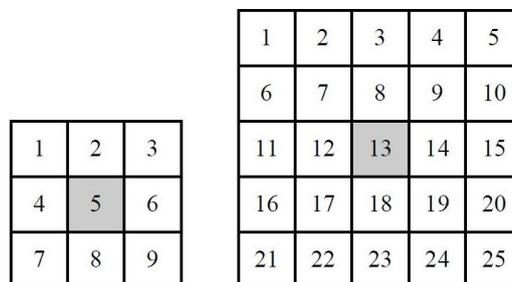


Figura 2.6: Diagramma di *array* di *pixel* usati per il calcolo degli indici di nitidezza. Cortesemente da [20].

Dati degli *array* di *pixel* 3×3 o 5×5 come mostrati in Figura 2.6, l'indice di nitidezza (SI) può essere calcolato usando un semplice algoritmo suggerito

da Ryall:

$$SI = abs[0.7071 * ((5 - 1) + (5 - 3) + (5 - 7) + (5 - 9)) + (5 - 2) + (5 - 4) + (5 - 6) + (5 - 8)] \quad (2.13)$$

Questo indice pesa i secondi vicini allo stesso modo dei primi vicini e somma i valori delle differenze [20].

L'algoritmo descritto da Goldsmith non permette di variare la grandezza della matrice utilizzata per calcolare l'indice di nitidezza. Gli autori del software utilizzato hanno modificato il metodo permettendo di variare la dimensione della matrice attraverso l'implementazione del seguente codice:

```
for (int y = vc; y < height - vc; y++) {
    for (int x = uc; x < width - uc; x++) {
        float sum = 0;
        for (int v = -vc; v <= vc; v++) {
            offset = x + (y + v) * width;
            for (int u = -uc; u <= uc; u++) {
                sum += Math.abs(pixels2[offset + u] - pixels2[x + y * width]);
            }
        }
        if (sharpness[x + y * width] < sum) {
            sharpness[x + y * width] = sum;
            depthimage[x + y * width] = (float) (s * distance);
            pixels[x + y * width] = (float) pixels2[x + y * width];
        }
    }
}
```

dove vc e uc rappresentano le coordinate del centro della regione di interesse sopra l'immagine originale; sum è l'indice di nitidezza, $height$ e $width$ sono l'altezza e la larghezza dell'immagine originale; $pixels2[offset + u]$ rappresenta i valori di intensità dei *pixel* vicini al centro della regione, $pixels2[x + y * width]$ è il valore del centro della regione studiata. La dimensione è definita dall'utente attraverso l'inserimento del valore voluto in una casella presente nell'interfaccia grafica. In particolare, come si vede dalla Figura 2.7, il parametro su cui bisogna agire è il *kernel size*, che decide la grandezza della regione d'interesse su cui andare a calcolare lo *sharpness index*.

Nella Figura 2.8 è mostrato un esempio di mappa delle profondità.

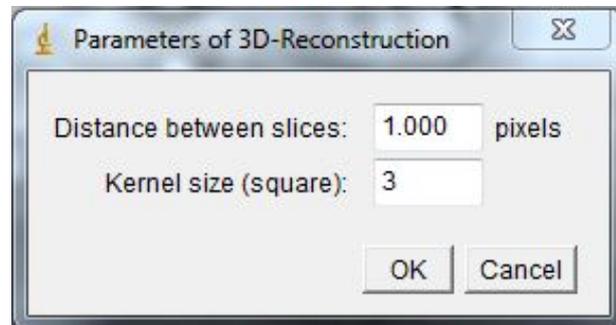


Figura 2.7: Interfaccia grafica di ImageJ del metodo Sharpness Index.

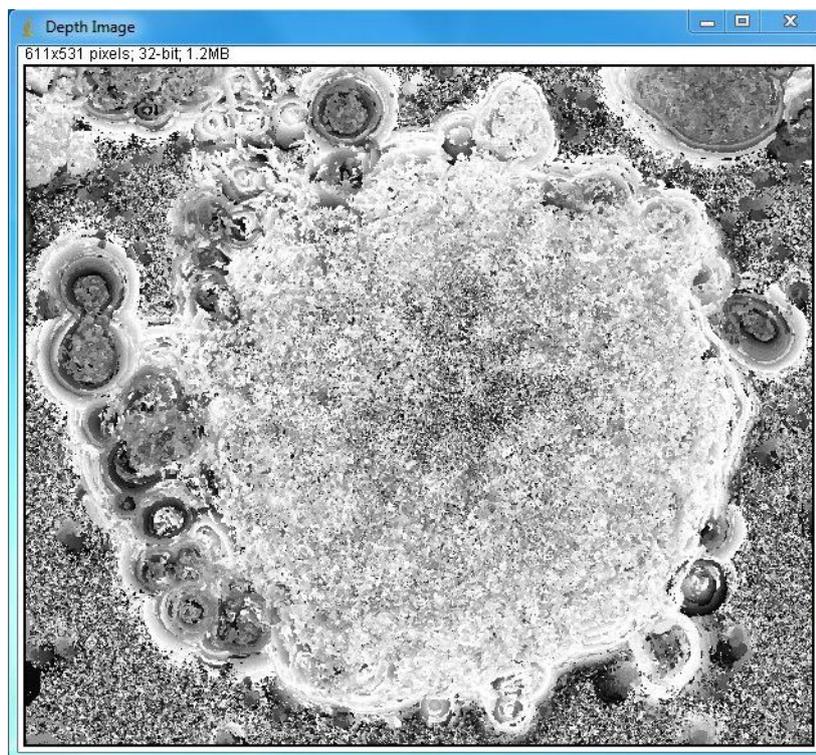


Figura 2.8: Immagine delle profondità (*Depth Image*) generata dal metodo.

2.5 Metodi nel dominio frequenziale

2.5.1 Complex Wavelet Transform

Implementato dal gruppo di ricerca *Biomedical Imaging Group* (BIG) dell'*École Polytechnique Fédérale De Lausanne* (EPFL) come *plug-in* di ImageJ. Il metodo proposto ricostruisce l'immagine composita finale partendo dallo sfruttare le aree a maggior salienza delle immagini di *input*. L'assunzione alla base del metodo è che le aree a fuoco presentano un elevato contenuto di dettagli rispetto alle aree non a fuoco, e queste possono essere facilmente individuate attraverso una ricerca locale di segnali ad alta frequenza. In particolare, la CWT risponde a pieno a questa esigenza. Il *software* presenta anche altri metodi per la messa a fuoco dell'immagine: Sobel, che sfrutta l'operatore di Sobel (Equazioni (2.9), (2.10), (2.11)) ed essenzialmente è un *edge detector*, con un piccolo filtraggio incorporato, che misura la forza dei contorni per ogni *pixel*, la varianza, le wavelet reali e le già citate wavelet complesse. L'analisi è stata fatta solo utilizzando la CWT perché è dichiarato dagli autori che, utilizzando l'approccio da loro proposto risulta essere il metodo in grado di fornire le maggiori prestazioni. Il programma presenta due interfacce distinte: una per i principianti (*Easy Mode*, Figura 2.9) in cui è possibile scegliere il compromesso tra qualità dell'immagine e velocità del processo, se si vuole visualizzare la mappa delle altezze e se su quest'ultima si vuole applicare un desiderato livello di filtraggio. La mappa delle altezze, come per gli altri metodi analizzati, è un'immagine in cui ogni *pixel* corrisponde al numero della *slice* nello *stack* da cui è stato preso il *pixel* utilizzato per la ricostruzione dell'immagine a fuoco. La seconda interfaccia è quella invece pensata per gli utenti più esperti (*Expert Mode*, Figura 2.10) in cui si ha molto più controllo sui parametri. Innanzitutto è possibile scegliere manualmente il metodo che si vuole utilizzare, inoltre si può decidere se si vuole fare una post-elaborazione sull'immagine finale e se si vuole visualizzare la mappa delle altezze con la possibilità di scegliere il tipo di filtro e anche la sua dimensione. Per il metodo da me analizzato, nella modalità *expert mode*, è anche possibile scegliere la dimensione dei banchi di filtri utilizzati per ottenere la CWT e anche il numero di decomposizioni che si vogliono effettuare. Il numero di decomposizioni influenza il livello di risoluzione che si vuole raggiungere, infatti, l'immagine al livello successivo, è ottenuta attraverso la decimazione dell'immagine al livello precedente, per cui il passaggio da un livello al suo successivo comporta la perdita di

dettaglio nell'immagine. Il diagramma di flusso della procedura è mostrato nel diagramma a blocchi della Figura 2.11.

Consiste di 5 passi fondamentali:

1. conversione da vettore a scalare: se le immagini in ingresso sono multicanale vengono convertite in immagini ad un solo canale
2. utilizzo della CWT,
3. applicazione delle regole di fusione e dei controlli di consistenza,
4. utilizzo della CWT inversa,
5. conversione da scalare a vettore: se necessario viene effettuata una riassegnazione per ottenere un'immagine multicanale.

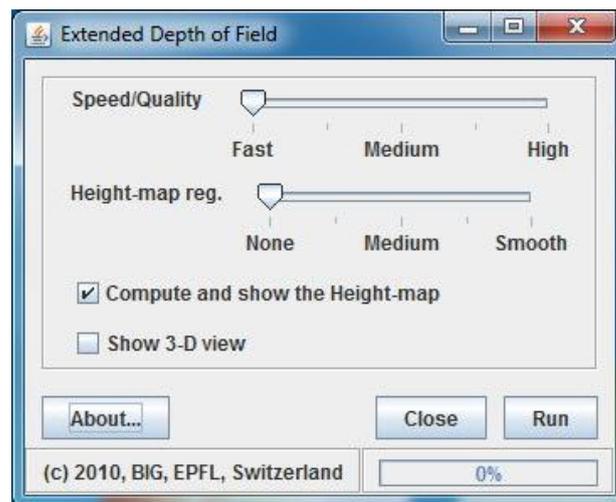


Figura 2.9: Interfaccia *Easy Mode*.

La trasformata wavelet può fornire alcune utili informazioni sul segnale analizzato. Per esempio, la presenza di coefficienti wavelet molto grandi, vicini a un determinato punto, serve come indicazione della presenza di dettagli notevoli in quel punto e, nel caso della fusione delle immagini è caratteristica di un'area a fuoco dell'immagine presa in analisi. Le wavelet usate per l'implementazione della CWT sono wavelet biortogonali o ortonormali, ciò comporta che l'informazione nelle diverse orientazioni è unica e non ridondante. In particolare sono state scelte le wavelet complesse di Daubechies. La regola di fusione consiste nel selezionare per ogni *pixel* la

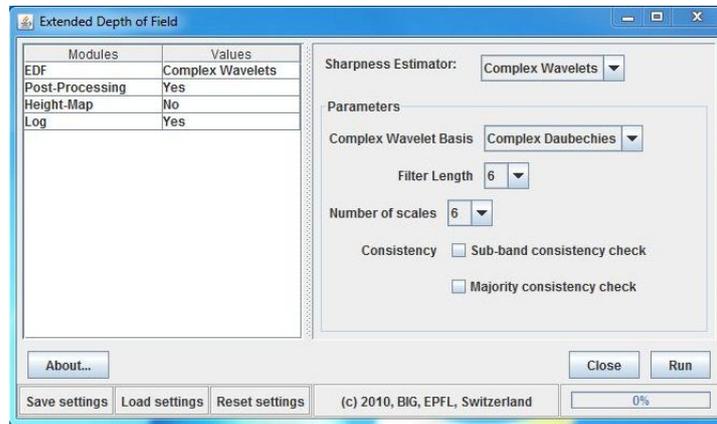


Figura 2.10: Interfaccia *Expert Mode*.

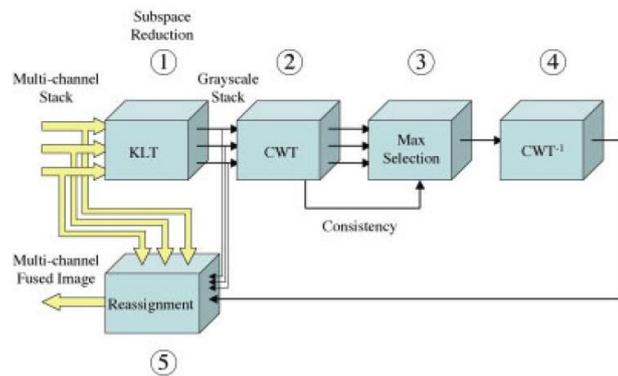


Figura 2.11: Diagramma di flusso della procedura. Cortesemente da [17].

slice con il valore assoluto del coefficiente wavelet più grande:

$$d_j(n, m) = c_j(n, m; \arg \max_z |c_j(n, m, z)|) \quad (2.14)$$

Sulla matrice dei coefficienti wavelet, creata attraverso le regole di fusione, si posso applicare dei controlli di consistenza, in questo caso specifico definiti controllo di consistenza sulle orientazioni (o sottobande) e controllo di consistenza spaziale. Il controllo di consistenza sulle sottobande (in inglese *subbands consistency check*) analizza le tre diverse orientazioni (vedi Figura A.3) in cui la trasformata wavelet scompone l'immagine per ogni scala e, se due coefficienti su tre sono attribuiti alla stessa *slice*, allora anche il terzo coefficiente è preso dalla medesima *slice*. Il controllo di consistenza spaziale (in inglese *spatial consistency check*) è svolto sulla mappa delle altezze (M) e utilizza una finestra di dimensione 3×3 . Se la maggioranza dei pixel della finestra proviene dalla k -esima *slice* mentre il *pixel* centrale è preso dalla j -esima, allora il numero $M(n, m) = j$ è cambiato in $M(n, m) = k$ e il rispettivo coefficiente è adattato: $d_{j,new}(n, m) = c_j(n, m; k)$. A causa della regola di fusione basata sulla scelta del massimo valore assoluto dei coefficienti, circa il 30% dei *pixel* sono sopra o sotto il *range* dinamico originale delle immagini. Tale incremento del *range* dinamico aumenta l'energia globale dell'immagine, il livello del rumore e la quantità di *pixel* saturati. Per evitare questi errori si può applicare sull'immagine finale una post-elaborazione definita come algoritmo di riassegnazione che seleziona il valore disponibile più vicino secondo i dati in scala di grigi [17]:

$$q(x, y) = s(x, y; \arg \min_z |p(x, y) - s(x, y; z)|) \quad (2.15)$$

2.5.2 Dual-Tree Complex Wavelet Transform

Il metodo DT-CWT è stato implementato da me per la piattaforma di calcolo MATLAB ©, sfruttando i codici, per eseguire la DT-CWT e la sua inversa, disponibili sul sito <http://eeweb.poly.edu/iselesni/WaveletSoftware/dt2D.html> dell'università politecnica di Brooklyn creati da Shihua Cai, Keyong Li, Ed il Prof. Ivan Selesnick. Come descritto nell'Appendice A, per ottenere la perfetta ricostruzione del segnale, il primo stadio di filtri deve essere diverso dai successivi. Nei codici scaricati, per ottenere la DT-CWT, si utilizza come banco di filtri per il primo stadio, dei filtri progettati dal Dr. Farras Abdelnour, *postdoc* presso l'università di Brooklyn (Figura 2.12(a), Figura 2.12(b)), mentre per gli stadi successivi si utilizzano i filtri disegnati

dal Prof. Nick Kingsbury dell'università di Cambridge (Figura 2.12(c), Figura 2.12(d)) [24].

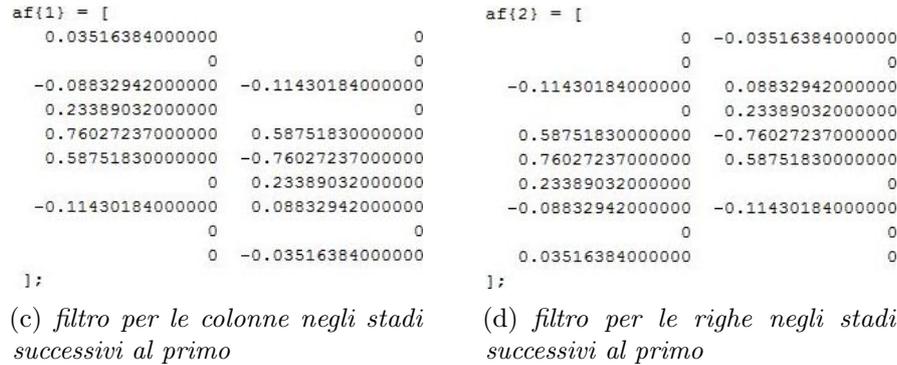


Figura 2.12: Per entrambi, la prima colonna rappresenta il filtro passa-basso, mentre la seconda rappresenta il filtro passa-alto.

I filtri di sintesi (per la ricostruzione) dell'immagine sono semplicemente i filtri di analisi della Figura 2.12 invertiti nel tempo.

L'implementazione dell'algoritmo di fusione delle immagini da me svolta, controlla come prima cosa che le righe e le colonne dell'immagine siano una potenza di due, se non viene soddisfatta questa condizione si effettua un *padding* con degli zero sull'immagine per portare il numero delle righe e delle colonne alla potenza di due più vicina. In seguito si fa in modo che il numero di livelli di decomposizione J dell'immagine non sia più fisso ma vari in base alle dimensioni dell'immagine:

$$J = \text{floor}(\log_2(\min(n.\text{righe}, n.\text{colonne}) - 3)) \quad (2.16)$$

Dove *floor* è una funzione di MATLAB © che approssima per difetto al numero intero più vicino. Quindi si applica la trasformata DT-CWT (Figura 2.13).

Per le regole di fusione ho letto diversi articoli [17], [46], [28], [40], [41], ma quello a cui mi sono particolarmente ispirato è l'articolo di Li *et al.* [29]. In quest'ultimo articolo gli autori fanno notare come in generale, le caratteristiche utili di un'immagine sono più grandi di un singolo *pixel*, per cui le regole di fusione applicate *pixel* per *pixel* potrebbero non essere il metodo migliore. Partendo da questa considerazione, per l'elaborazione delle sottobande ad alta frequenza, ho utilizzato una finestra di dimensioni 3×3 che scansioni tutta l'immagine; ad ogni passo, per tutte le *slice* che compongono lo *stack*, è fatta la media dei valori assoluti dei coefficienti wavelet all'interno della finestra. La media serve come misuratore del livello di attività, infatti come citato nell'articolo [17], la presenza di coefficienti wavelet grandi nell'intorno di un punto è indice di aree a fuoco dell'immagine. In base alle considerazioni appena fatte, la *slice* che avrà la media più alta all'interno della finestra sarà presumibilmente anche la *slice* dello *stack* che rappresenterà quell'area delle immagini nella maniera più nitida. Il coefficiente al centro della finestra nella *slice* selezionata, rappresenterà, nella stessa posizione, il coefficiente per l'immagine finale. Per le basse frequenze, i coefficienti da utilizzare per la ricostruzione dell'immagine a fuoco sono ottenuti facendo una media dei coefficienti delle immagini ottenute attraverso i filtri passa-basso. A questo punto si esegue l'antitrasformata e si ottiene l'immagine completamente a fuoco che però generalmente presenta un *range* dinamico maggiore delle immagini originali e questo può portare a un aumento del livello di rumore e della saturazione dei *pixel* [17]. Per evitare questo problema, l'algoritmo esegue una post-elaborazione dell'immagine utilizzando il modello proposto nell'articolo [6]. Si calcolano il massimo e il minimo assoluto lungo tutto lo *stack* per ogni posizione:

$$q_{min} = \arg \min_z s(x, y; z) \quad (2.17)$$

$$q_{max} = \arg \max_z s(x, y; z) \quad (2.18)$$

L'immagine finale è quindi processata per assicurarsi che il valore dei

suoi *pixel* cada all'interno dello stesso *range* delle immagini originali.

$$p(x, y) = \begin{cases} q_{max}(x, y) & \text{if } p(x, y) > q_{max} \\ q_{min}(x, y) & \text{if } p(x, y) < q_{min} \\ p(x, y) & \text{otherwise} \end{cases} \quad (2.19)$$

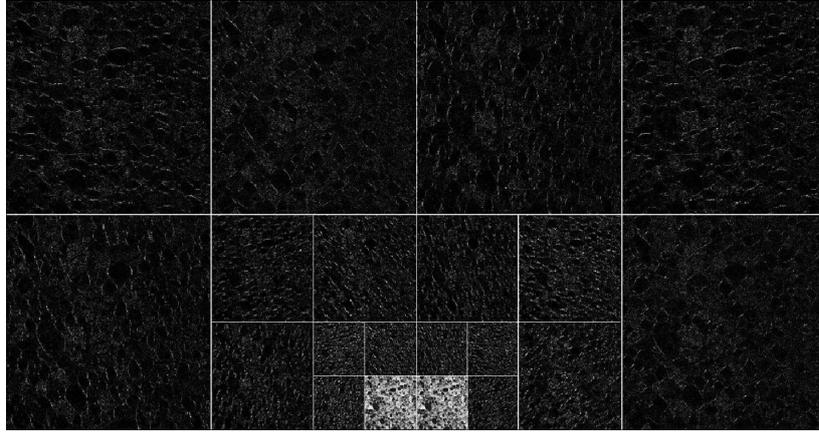


Figura 2.13: Rappresentazione di una decomposizione a 3 livelli tramite DT-CWT.

2.5.3 Curvelet Transform

Come per il metodo DT-CWT, per la realizzazione della CT ho sfruttato il *Curvelet Toolbox* scaricabile gratuitamente dal sito www.curvelet.org e sviluppato da Emmanuel Candes, Laurent Demanet, David Donoho e Lexing Ying che sono gli stessi autori della teoria della CT.

La CT è realizzata tramite l'approccio di *wrapping* (termine inglese che tradotto significa: avvolgimento, impacchettamento). Quest'approccio considera una suddivisione del piano delle frequenze come in Figura A.6 ma con una differenza: invece che usare una griglia inclinata, utilizza una griglia rettangolare regolare.

Il punto cruciale di quest'approccio è il seguente: nel piano delle frequenze, la finestra di localizzazione discreta $\tilde{U}[n_1, n_2]$ non si adatta perfettamente a una finestra rettangolare di dimensioni $\sim 2^j \times 2^{j/2}$, allineata con gli assi, in cui può essere applicata la trasformata di Fourier inversa 2D (in inglese *inverse fast Fourier transform*, 2D IFFT) per calcolare i coefficienti curvelet. L'idea alla base dell'approccio *wrapping* è il considerare la periodizzazione dei campioni nel dominio delle frequenze e "impacchettare" questi dati in una finestra rettangolare nell'intorno dell'origine degli assi. Si considera $P_{j,l}$ un

parallelogramma che contiene la finestra di localizzazione discreta $\tilde{U}[n_1, n_2]$. Per ogni scala j , esistono due costanti $L_{1,j} \sim 2^j$ e $L_{2,j} \sim 2^{j/2}$ che, per ogni orientazione θ_l permettono di suddividere lo spazio bidimensionale delle frequenze attraverso la traslazione di $P_{j,l}$ di multipli di $L_{1,j}$ nella direzione orizzontale e $L_{2,j}$ nella direzione verticale. La corrispondente periodizzazione dei dati d è data da:

$$Wd[n_1, n_2] = \sum_{m_1 \in \mathbb{Z}} \sum_{m_2 \in \mathbb{Z}} d[n_1 + m_1 L_{1,j}, n_2 + m_2 L_{2,j}] \quad (2.20)$$

I dati “impacchettati”, nell’intorno dell’origine, sono definiti come la limitazione di $Wd[n_1, n_2]$ per gli indici n_1, n_2 all’interno di un rettangolo di dimensioni $L_{1,j} \times L_{2,j}$ vicino all’origine (Figura 2.14):

$$0 \leq n_1 < L_{1,j}, 0 \leq n_2 < L_{2,j} \quad (2.21)$$

La trasformazione di “impacchettamento” è una semplice reindicizzazione dei dati [10].

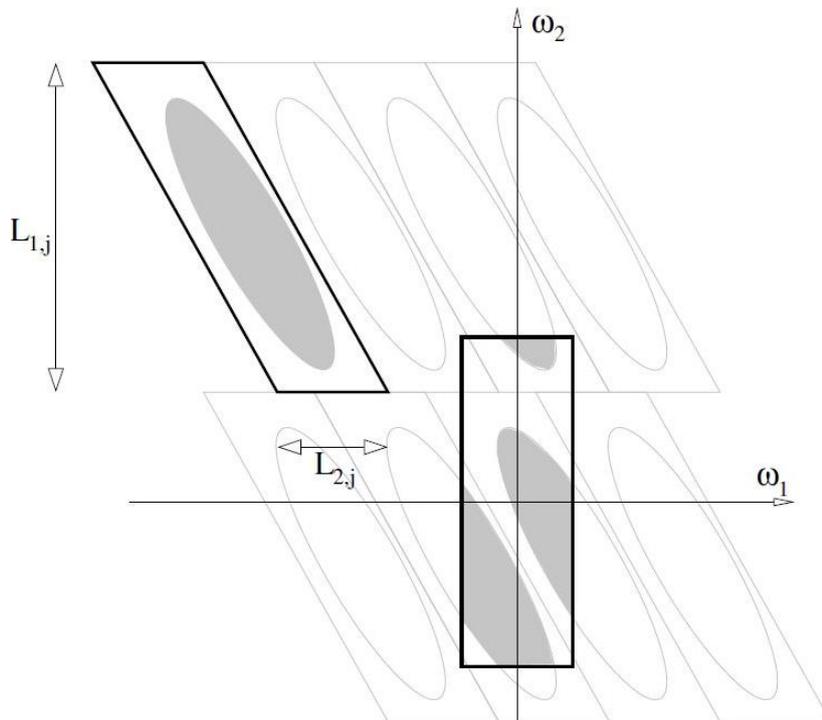


Figura 2.14: Impacchettamento dei dati, inizialmente all’interno di un parallelogramma, in un rettangolo attraverso la periodizzazione. Cortesemente da [10].

All’interno del *Curvelet Toolbox*, la CT realizzata è leggermente diversa da

quella descritta nell'Appendice A , perché nel pacchetto è stata costruita la trasformata curvelet estesa specularmente (in inglese *extended-mirror curvelet transform*, EM-CT). Questa trasformata tratta i contorni dell'immagine attraverso un'estensione speculare, mentre la trasformata standard li tratta attraverso una periodizzazione. Si considera una funzione reale $u(X)$ con $X = (x_1, x_2)$ e si definisce la sua estensione speculare come:

$$\tilde{u}(x_1, x_2) = u(|x_1|, |x_2|) \quad (2.22)$$

posta in $[-1, 1]^2$. La linea guida da seguire per la costruzione della ME-CT è di lavorare sull'estensione \tilde{u} , e di eliminare i calcoli ridondanti dove possibile. Il primo passo per realizzare la ME-CT è di calcolare la trasformata del coseno di u al posto della trasformata di Fourier di \tilde{u} . Si può dimostrare che la trasformata di Fourier di $\tilde{u} = F\{\tilde{u}(X)\} = \hat{\tilde{u}}$ corrisponde alla trasformata del coseno della funzione originale u , moltiplicata per un fattore 4. Come conseguenza di questa ridondanza, il calcolo di $\hat{\tilde{u}}(\omega_1, \omega_2)$ può essere limitato solo al primo quadrante con $\omega_1, \omega_2 > 0$, e questo è esattamente ciò che fa la trasformata del coseno. Il secondo passo della ME-CT consiste nell'“impacchettare” i dati ottenuti dall'estensione speculare della trasformata del coseno, in finestre nell'intorno dell'origine. La simmetria dei dati \hat{f} , oltre alla simmetria dell'architettura della trasformata, può essere utilizzata per identificare le ridondanze. Come risultato di queste simmetrie si possono considerare solo gli angoli $0 < \theta_l < \pi/2$, e ottenere tutti gli altri attraverso rotazioni di $\pm\pi/2$.

Le regole di fusione per realizzare l'algoritmo le ho implementate ispirandomi all'articolo [45]. Innanzitutto bisogna processare in maniera separata le sottobande ad alta frequenza e l'immagine passa-basso perché avendo un contenuto informativo differente devono essere elaborate con regole di fusione differenti. Si è proceduto nel seguente modo:

- Per le sottobande ad alta frequenza, i coefficienti curvelet maggiori corrispondono alle caratteristiche dell'immagine ad alta risoluzione e si assume che queste caratteristiche siano in una regione a fuoco dell'immagine. Perciò, selezionando attraverso lo *stack* i coefficienti con il valore assoluto più alto per ogni posizione, orientazione e scala, si garantisce che le caratteristiche salienti dell'immagine siano preservate.
- Per l'immagine passa-basso si è utilizzato un approccio basato sull'assunzione della consistenza tra sottobande. Ipotizzando ciò, i coefficienti

curvelet corrispondenti ad una determinata caratteristica dell'immagine, possono essere presi dalla stessa *slice* indipendentemente dal livello di scala e dalla orientazione. Questo significa che il coefficiente curvelet, per ogni posizione (x, y) nell'immagine passa-basso può essere preso dalla stessa *slice* dalla quale è stato preso il relativo coefficiente curvelet per le alte frequenze. Come approssimazione, si può selezionare come immagine passa-basso la *slice* da cui sono stati presi la maggioranza dei corrispondenti coefficienti.

L'immagine risultante è stata infine *post* elaborata, come nel metodo DT-CWT, sfruttando le Equazioni (2.17), (2.18), (2.19).

Capitolo 3

Nuovo metodo per l'estensione della profondità di fuoco

Questo nuovo metodo per l'estensione della profondità di fuoco, basato su un'analisi puntuale locale (da qui in seguito denominato *AnPuLo*), seguendo la classificazione utilizzata nel Capitolo 2, è inserito nella classe dei metodi che lavorano nel dominio spaziale. L'algoritmo è basato su un'analisi puntuale-locale delle varie immagini originali che compongono lo *stack* di partenza: i *pixel* $p(x, y)$ delle singole immagini sono comparati tra loro analizzando i vari vettori z che compongono lo *stack*. Attraverso una regola di fusione, fondamentalmente basata sulla ricerca del massimo valore in z della derivata locale, viene ricostruita l'immagine composta finale copiando i valori dei *pixel* dalle immagini originali. L'algoritmo è stato da me implementato ispirandomi all'articolo di Valdecasas *et al.* [46] nel quale è spiegato come gli operatori che permettono il riconoscimento dei contorni (in inglese *edges detector*) posso essere usati per la ricostruzione di immagini 2-D completamente a fuoco, sfruttando l'assunzione che generalmente tra più rappresentazioni della stessa scena si considera a "maggior fuoco" l'immagine che contiene il maggior numero di dettagli. L'algoritmo è stato completamente implementato in codice MATLAB (© The MathWorks, Inc., Massachusetts, USA)(vedi Appendice B.1, B.2).

3.1 Schema a blocchi e descrizione

In Figura 3.1 è riportato lo schema a blocchi che illustra schematicamente la logica dell'algoritmo creato:

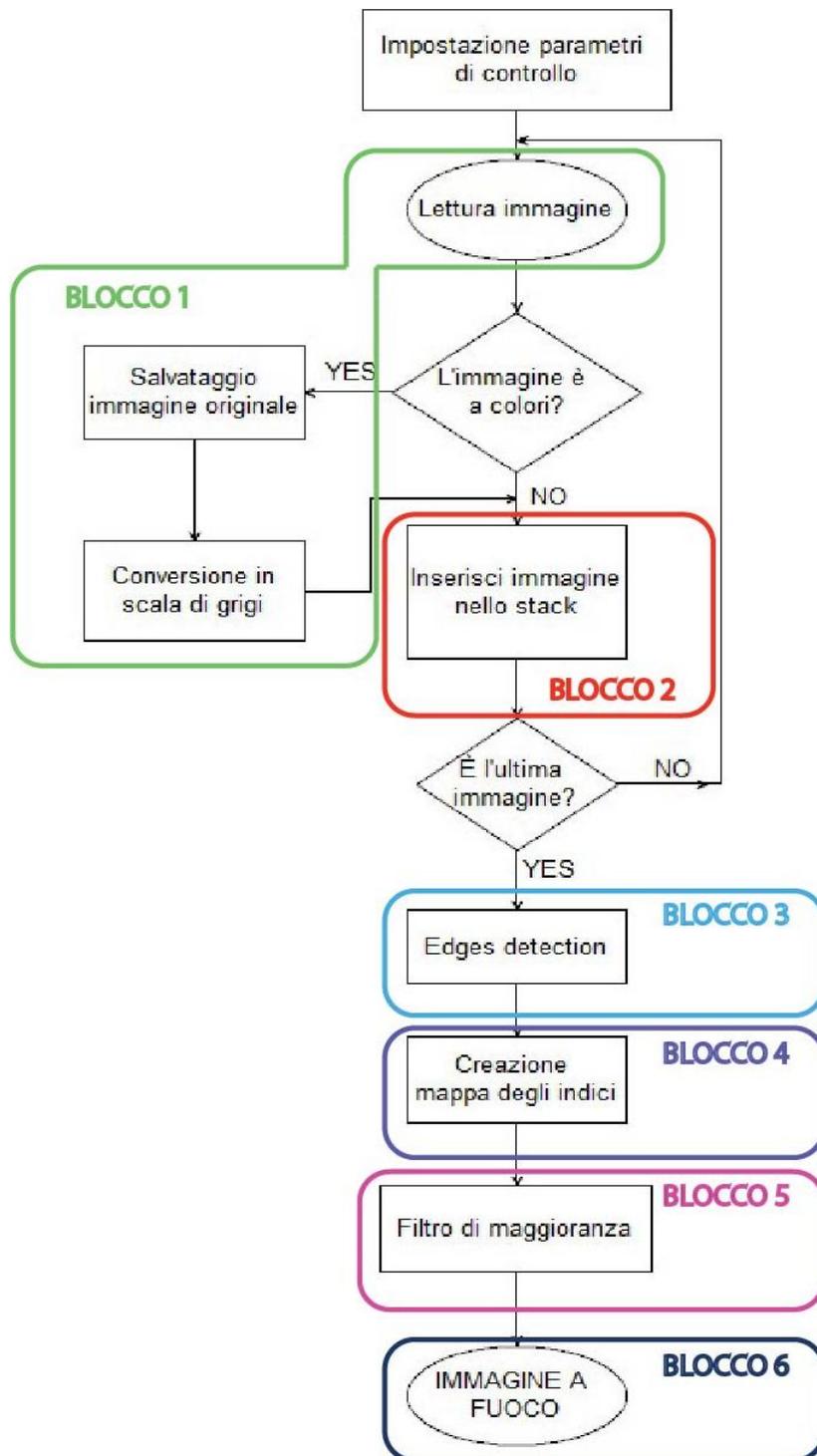


Figura 3.1: Schema a blocchi dell'algoritmo AnPuLo.

Descriviamo sinteticamente per blocchi il funzionamento del programma:

BLOCCO 1: Lettura e conversione delle immagini. All'inizio dell'algoritmo vi è un modulo creato per leggere le immagini e convertirle in scala di grigi attraverso l'utilizzo dell'analisi alle componenti principali (in inglese *principal component analysis*, PCA) [17].

BLOCCO 2: Creazione dello *stack* di immagini. Le immagini prese a differenti piani focali vengono "impilate" una sull'altra fino a formare un'array a tre dimensioni (lo *stack* d'immagini) che ha lo stesso numero di righe e di colonne delle immagini di partenza e il numero di canali pari al numero delle immagini N che sono state "impilate" (dimensioni della matrice $n \times m \times N$).

BLOCCO 3: Rilevamento degli *edge*. In questo blocco per ogni immagine dello *stack* vengono rilevati i contorni attraverso l'utilizzo della derivata locale alle differenze centrali.

BLOCCO 4: Creazione mappa degli indici (denominata anche mappa delle altezze o delle profondità). È una matrice 2-D di dimensioni $n \times m$, in cui ogni *pixel* corrisponde all'indice, lungo l'asse z , dell'immagine dello *stack* dalla quale è stato preso quel *pixel* per la creazione dell'immagine composta finale.

BLOCCO 5: Filtro di maggioranza [29], [17]. È una tecnica di filtraggio a finestra mobile. In particolare è stata utilizzata una finestra quadrata 5×5 per analizzare *pixel* per *pixel* la mappa degli indici. Se più della metà dei *pixel* contenuti all'interno della finestra hanno lo stesso valore (quindi provengono dalla stessa immagine), allora anche al *pixel* centrale viene assegnato quel valore.

BLOCCO 6: Immagine composta finale di output. Basandosi sulla mappa di profondità creata per ogni posizione x - y viene copiato il valore

da una delle immagini originali che compongono lo *stack*. In fine, viene visualizzata in *output* l'immagine finale completamente a fuoco dell'oggetto in analisi.

3.2 Conversione in scala di grigi attraverso la PCA e formazione dello *stack* di immagini

Il primo passo dell'algoritmo consiste nella lettura delle singole immagini che andranno a formare lo *stack*. Se le immagini originali sono già in scala di grigi, l'algoritmo non fa altro che creare una matrice *Stack* di dimensioni $n \times m \times N$. di immagini e "impilare" le immagini che saranno processate per ricostruire l'immagine composita finale. Invece, se le immagini originali sono in formato multi-canale, ad esempio a colori RGB, quello che ad esempio si potrebbe pensare di fare sarebbe di utilizzare l'algoritmo in maniera indipendente su ogni singolo canale dell'immagine. Questo è sicuramente un approccio plausibile e corretto ma occorrerebbe eseguire l'intero algoritmo diverse volte (n . di immagini \times n . di canali) aumentando notevolmente il costo computazionale e lo spazio necessario alla memorizzazione dei dati. Perciò, prendendo spunto dall'articolo di Foster *et al.* [17], nel nostro algoritmo abbiamo scelto di pre-convertire le immagini multi-canale in immagini in scala di grigi, in modo da eseguire l'algoritmo una sola volta per ogni immagine dello *stack*. Le immagini originali a colori vengono comunque archiviate perché sono richiamate al termine del processamento per creare eventualmente anche una versione a colori dell'immagine finale. Il metodo tradizionale per eseguire la conversione da immagini a colori nello spazio RGB a immagini in scala di grigi consiste nel calcolare la luminanza attraverso una media pesata (in cui i pesi sono fissi) dei tre canali (*Red*, *Green*, *Blue*) di partenza:

$$GRAY = 0.30Red + 0.59Green + 0.11Blue \quad (3.1)$$

Questa è la tipica trasformazione universale da colori in scala di grigi che è stata ottimizzata per le immagini naturali. Tuttavia, nelle immagini microscopiche, un particolare colore può essere maggiormente presente, oppure, posso essere disponibili diversi tipi di canali rispetto ai tradizionali RGB (es. immagini in fluorescenza). Perciò si è scelto di applicare una conversione da vettore-a-scalare (in inglese *vector-to-scalar conversion*) delle

immagini di partenza:

$$\{s^{(k)}(x, y; z)\}_z \rightarrow \{s(x, y; z)\} \quad (3.2)$$

dove $s(x, y; z) = \sum_k w_k s^{(k)}(x, y; z)$ è una combinazione lineare dei dati provenienti dai k canali differenti. I pesi sono ottenuti dalla PCA e corrispondono alle direzioni che hanno massima varianza nello spazio multi-canale. La PCA è una tecnica statistica frequentemente usata nell'elaborazione dei segnali per ridurre la dimensione dei dati o per rendere scorrelati un insieme di dati originariamente correlati. Nel caso specifico, viene usata per ridurre la dimensione di immagini a colori, infatti da immagini a multi-canale si ottiene un'immagine a singolo canale contenente la maggior parte dell'informazione. In particolare, si scompone l'immagine nei singoli canali, per ogni canale viene fatta la PCA, con la trasformata di *Karhunen-Loève* (KLT), e viene preso come peso l'autovalore relativo all'autovettore che corrisponde alla direzione di massima varianza. In questo modo, un colore predominante sarà pesato maggiormente durante l'algoritmo di fusione. Le immagini in scala di grigi così create preservano una maggiore salienza e contrasto rispetto a quelle ottenute con la conversione tradizionale [17].

3.3 *Edge detection*

Tipicamente, tutti gli approcci nel dominio spaziale sono basati su due passi fondamentali: il primo consiste nel rilevamento delle regioni a fuoco delle immagini, il secondo prevede il processo di fusione delle informazioni appena rilevate per ottenere l'immagine completamente a fuoco. Generalmente per capire quali zone delle immagini sono a fuoco si utilizzano degli operatori che realizzano il rilevamento dei contorni. Molto comunemente, in letteratura, a questo scopo è utilizzato l'operatore di Sobel che oltre a rilevare i contorni, esegue contemporaneamente anche un leggero filtraggio [46]. Nell'algoritmo proposto, per semplificare si è deciso di separare i due stadi. Come *edges detector* si è scelto la via più semplice: la derivata, in x e in y , approssimata con una differenza finita centrale (molto più robusta rispetto alla differenza finita in avanti e alla differenza finita all'indietro). Per velocizzare l'algoritmo, si è deciso di calcolare separatamente la derivata in x e la derivata in y utilizzando le seguenti maschere:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$$G_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad (3.4)$$

E ottenere la derivata x - y approssimata semplicemente sommando localmente il valore assoluto delle due singole derivate. In seguito, sullo *stack* d'immagini in cui sono stati rilevati i contorni, è applicato un filtro a media per eliminare picchi di alte derivate dati ad esempio da rumore e per propagare l'influenza di una zona ad elevato contrasto seguendo l'assunzione che zone contigue saranno a fuoco in immagini contigue.

3.4 Mappa degli indici e filtro di maggioranza

Per combinare le regioni a fuoco per formare l'immagine composta finale, generalmente si utilizza la regola di selezione del massimo o del minimo degli *edges* per ogni *pixel* lungo tutto lo *stack* d'immagini [46]. Nell'algoritmo proposto, non ci si limita a utilizzare la regola di selezione del massimo valore della derivata per ogni posizione x - y lungo lo *stack*: prima si costruisce una mappa degli indici (o *depth map*, Figura 3.2) che riporta per ogni *pixel* il numero della *slice*, appartenente allo *stack* delle derivate, che contiene il valore massimo della derivata lungo z e, in seguito, sulla *depth map*, è applicato un filtro di maggioranza (in inglese *majority filter*) [29], [17]. Il filtro di maggioranza utilizzato è costituito da una finestra mobile di dimensioni 5×5 in cui si controlla se più della metà dei *pixel* contenuti all'interno della finestra hanno lo stesso valore (cioè provengono dalla stessa *slice*). In tal caso anche al *pixel* centrale viene assegnato quel valore. Il filtro di maggioranza serve per dare maggior continuità alla mappa degli indici e per garantire che le caratteristiche dominanti siano incorporate, le più complete possibili, all'interno della nuova immagine.

3.5 Creazione dell'immagine finale

L'immagine finale composta è costruita sfruttando la *depth map* e copiando, per ogni posizione x - y della nuova immagine, il valore originale del *pixel* dall'immagine specifica indicata dalla mappa.

Se le immagini di partenza sono a colori, e si vuole ottenere anche in uscita un'immagine a colori, allora l'algoritmo crea una matrice di zeri di dimensioni pari a $n \times m \times 3$. Sfruttando la *depth map* e le immagini iniziali

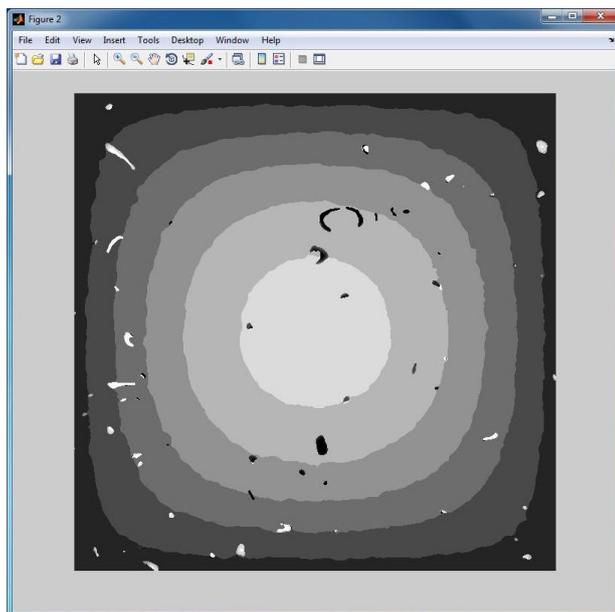


Figura 3.2: Mappa degli indici o *depth map* utilizzata per la ricostruzione dell'immagine finale composta.

archivate durante l'esecuzione del primo blocco dell'algoritmo, la matrice precedentemente creata, viene riempita copiando per ogni *pixel* il vettore RGB dell'immagine originale a colori dalla *slice* dello *stack* indicata dalla mappa per quella posizione.

Capitolo 4

Metriche di valutazione

Una delle problematiche principali che interessa il campo di ricerca dell'estensione della profondità di campo è l'impossibilità di definire quale metodo ottenga il risultato migliore perché, ancora oggi, non esiste una metrica testata e un approccio validato per poter confrontare i diversi metodi, in particolare quando si utilizzano immagini reali per le quali è molto difficile, se non impossibile, avere la verità assoluta (in inglese *ground truth*). Per cui, nei casi reali, tipicamente viene effettuata solamente un'analisi qualitativa visiva, che però risulta essere affetta da una grandissima variabilità e soggettività. Risulta quindi arduo considerare questo tipo di analisi uno strumento per la classificazione di differenti algoritmi in esame. Un approccio alternativo di valutazione, per le applicazioni dove le immagini “fuse” sono create in funzione di successivi *task*, quali ad esempio rilevamento, tracciamento e classificazione di oggetti, consiste nel valutare gli algoritmi di fusione in termini di miglioramento della prestazione finale. Questo approccio richiede un compito ben definito per il quale possono essere fatte misurazioni quantitative. Tuttavia, ciò richiede solitamente alti costi in termini di tempo e i risultati ottenuti spesso non sono direttamente dipendenti alla qualità del metodo di fusione utilizzato.

4.1 Metriche standard in *image processing*

Molte metriche sono state proposte in letteratura per valutare le immagini composite finali prodotte dai vari metodi di fusione. Una panoramica può essere trovata in [56]. Generalmente, le metriche maggiormente usate nell'elaborazione delle immagini sono l'errore quadratico medio (in inglese *Mean Squared Error*, MSE), il rapporto segnale-rumore (in inglese *Signal-to-*

Noise Ratio, SNR) e l'indice di qualità universale (in inglese *Universal Quality Index*, UQI) [48]. L'MSE e l'SNR sono metriche tipicamente usate nell'analisi di dati e segnali, ed in particolare per l'elaborazione delle immagini sono definite rispettivamente come:

$$MSE(g, f) = \frac{\sum_{x=1}^P [g(x) - f(x)]^2}{P} \quad (4.1)$$

$$SNR(g, f) = 10 \log_{10} \frac{\sum_{x=1}^P g(x)^2}{P * MSE(g, f)} \quad (4.2)$$

Dove $g(x)$ è la *ground truth* di riferimento, $f(x)$ è l'immagine che si vuole valutare, x rappresenta le coordinate del pixel e P è il numero totale di *pixel* che compongono le immagini $g(x)$ e $f(x)$. A differenza delle precedenti, l'UQI è una metrica utilizzata solo nella valutazione delle immagini, ed è definita come:

$$UQI(g, f) = \left(\frac{\sigma_{gf}}{\sigma_g \sigma_f} \right) \left(\frac{2\bar{x}_g \bar{x}_f}{\bar{x}_g^2 + \bar{x}_f^2} \right) \left(\frac{2\sigma_g \sigma_f}{\sigma_g^2 + \sigma_f^2} \right) \quad (4.3)$$

Dove \bar{x}_g , \bar{x}_f , σ_g , σ_f , σ_{gf} sono rispettivamente la media, la deviazione standard e la covarianza delle immagini $g(x)$ e $f(x)$. Il primo fattore nell'Equazione (4.3) misura il grado di correlazione lineare tra $g(x)$ e $f(x)$ e varia tra $[-1, 1]$, il secondo (Equazione (4.3)) corrisponde alla distorsione della luminanza e ha un *range* dinamico tra $[0, 1]$ e l'ultimo componente (Equazione (4.3)) misura quanto sono simili i contrasti tra le immagini e i suoi valori variano tra $[0, 1]$. Quindi l'UQI può variare tra $[-1, 1]$ e il valore massimo, cioè 1, è raggiunto quando $g(x)$ e $f(x)$ sono identiche [48]. L'UQI risulta essere un ottimo indice per valutare la qualità delle immagini e soprattutto è definito matematicamente senza l'impiego di modelli basati sul sistema visivo umano (in inglese *human visual system*, HSV); ciò rende l'UQI una misura indipendente dalle condizioni di osservazione e permette di ottenere un valore oggettivo sull'immagine analizzata.

4.2 Estensione dell'UQI per *image fusion*

Caratteristica comune delle tre metriche riportate nel Paragrafo 4.1 è che tutte richiedono la *ground truth* per essere in grado di valutare la qualità dell'immagine, ma come già detto, spesso nei problemi di fusione delle immagini questa è un'informazione mancante, in particolare quando si

utilizzano dei set di dati reali. Per superare questo problema, Piella *et al.* proposero in [37] un'estensione dell'UQI, conosciuta come “metrica di Piella” (in inglese *Piella's Metric*, PM), adatta per stimare, senza la *ground truth*, la qualità delle immagini “fuse”, ma con il grosso limite che può essere applicata solo a *stack* di massimo due immagini. La PM è stata estesa in [8] ad un'altra metrica, definita UQIN, adatta a lavorare non solo con due *slice* ma su *stack* composti da N immagini. Sia per la PM che per l'UQIN sono state proposte tre differenti versioni semplicemente introducendo differenti pesi ai termini dell'equazione base. La versione base per l'UQIN è stata così definita dagli autori:

$$UQIN(s_1, \dots, s_n, f) = \frac{1}{|T|} \sum_{w \in T} \sum_{n=1}^N \lambda_{s_n}(w) UQI(s_n, f|w) \quad (4.4)$$

Dove s_n è una delle N *slices* utilizzate per creare l'immagine composta finale (per PM $N = 2$), w è una piccola regione di interesse (in inglese *Region Of Interest*, ROI) dell'insieme T in cui le immagini sono state suddivise e $\lambda_{s_n}(w)$ è definita come:

$$\lambda_{s_n} = \frac{Y(s_n|w)}{\sum_{n=1}^N Y(s_n|w)} \quad (4.5)$$

Dove $Y(s_n|w)$ è la salienza locale dell'immagine s_n , tipicamente definita come la varianza locale all'interno della ROI w . Questa versione pesa in maniera uguale tutte le misure di qualità ottenute all'interno di ogni finestra. Questo è in contrasto con i canoni definiti per l'HSV, in cui si conferisce maggiore importanza alle regioni dell'immagine che visivamente risultano essere più salienti. Perciò è stata definita anche una variante dell'indice di qualità che pesa maggiormente le ROI dove la salienza delle immagini di ingresso è maggiore. La salienza complessiva di una ROI può essere ottenuta come:

$$C(s_1, \dots, s_n, f) = \max \{ \{Y(s_1|w), \dots, Y(s_n|w)\} \} \quad (4.6)$$

E la nuova versione pesata dell'UQIN (da qui in avanti denominata $UQIN_W$) può essere espressa come:

$$UQIN_W(s_1, \dots, s_n, f) = \sum_{w \in T} \sum_{n=1}^N c(w) \lambda_{s_n}(w) UQI(s_n, f|w) \quad (4.7)$$

Dove $c(w) = C(s_1, \dots, s_n|w) / \sum_{w \in T} C(s_1, \dots, s_n|w)$.

L'ultima modifica sull' $UQIN_W$ è stata introdotta per tenere conto di alcuni aspetti dell'HSV in cui le informazioni sui contorni della scena sono utilizzate per ottenere l'indice di qualità definito $UQIN_E$. In questo approccio le informazioni sui contorni sono usate nell'Equazione (4.7) al posto dell'immagini in scala di grigi. Considerando s_1^e, \dots, s_n^e, f^e le corrispondenti immagini dei contorni (ottenute semplicemente derivando) delle immagini s_1, \dots, s_n, f allora combinando $UQIN_W(s_1, \dots, s_n, f)$ e $UQIN_W(s_1^e, \dots, s_n^e, f^e)$ si ottiene il nuovo indice definito come:

$$UQIN_E(s_1, \dots, s_n, f) = UQIN_W(s_1, \dots, s_n, f)^{1-\alpha} UQIN_W(s_1^e, \dots, s_n^e, f^e)^\alpha \quad (4.8)$$

Dove α è il parametro che esprime il contributo delle immagini dei contorni rispetto alla immagini originali. Tutti gli indici appena descritti possono assumere valori compresi tra $[-1, 1]$ e l'immagine finale sarà quantitativamente migliore più il valore dell'indice risulta essere vicino a uno.

Capitolo 5

Materiali

La mia tesi si inserisce nel contesto di un progetto di collaborazione nata tra il Laboratorio di Bioscienze dell'Istituto scientifico Romagnolo per lo Studio e la cura dei Tumori (IRST) e il gruppo di ricerca *Computer Vision Group* (CVG) dell'Università di Bologna, volta principalmente alla caratterizzazione di aggregati multicellulari tramite l'impiego di tecniche di *image processing* applicate ad immagini acquisite con microscopi ottici tradizionali. Il progetto, denominato STAMINAL (caratterizzazione di cellule STAminali tramite supporto per l'analisi autoMatIca delle immagini microscopiche in terapia pre-clinica) riguarda principalmente lo studio morfologico e funzionale degli aggregati multicellulari a forma sferica (sferoidi) di cellule provenienti dai bronchi alveolari (broncosfere). In particolare, si vuole studiare le broncosfere formate da cellule polmonari CAEP (carcinoma squamoso) e RAL (adenocarcinoma) che si creano in particolari condizioni. Utilizzando appositi bioreattori gli sferoidi possono raggiungere anche una dimensione di circa 1mm di diametro. Considerando che la profondità di fuoco di un tradizionale microscopio ottico dipende dall'obiettivo utilizzato ma è ordini di grandezza inferiore, risulta impossibile mettere a fuoco gli sferoidi nella loro totalità. Partendo da questi limiti del microscopio, sono state acquisite sequenze di immagini a diversi piani focali (passo 5 μm) e successivamente si è cercato quindi di ricostruire le immagini 2-D finali completamente a fuoco.

5.1 Strumentazione di base di un laboratorio di colture cellulari

Il Laboratorio di Bioscienze è principalmente dedicato a studi clinici in ambito oncologico. Ciò nonostante risulta essere particolarmente attivo in studi di ricerca nei seguenti campi: diagnostica molecolare, farmacologia preclinica, marcatori predittivi, biologia cellulare, oste oncologia. I miei referenti sono il direttore e la responsabile del laboratorio, rispettivamente il Dott. Wainer Zoli e la Dott.ssa Anna Tesei. Teoricamente, un laboratorio di colture cellulari ideale dovrebbe essere delle dimensioni adeguate per potere accogliere tutto l'equipaggiamento necessario e dovrebbe essere isolato dalle altre stanze da almeno una porta. La collocazione perfetta sarebbe quella in cui tra esso e gli altri ambienti ci fosse almeno una stanza da usare come "spogliatoio", in cui cioè togliersi i camici da usare nel resto del laboratorio e indossare quello dedicato alle colture. L'atmosfera nella stanza delle colture dovrebbe essere in pressione positiva, cioè ad una pressione leggermente maggiore che non nel resto del laboratorio, in modo da prevenire l'ingresso di aria (potenzialmente contaminata) dagli ambienti limitrofi. Il pavimento dovrebbe essere in materiale non poroso, facile da pulire, di colore chiaro in modo da rivelare facilmente macchie dovute a residui caduti; le finestre dovrebbero essere grandi in modo da garantire una buona illuminazione con luce naturale, ma non devono mai essere aperte, per evitare di fare entrare aria esterna carica di contaminanti. A causa del riscaldamento causato dalle apparecchiature, sarebbe necessario potere controllare e mantenere la temperatura e l'umidità entro dei livelli definiti. Il Laboratorio di Farmacologia Preclinica dell'IRST ha una dimensione che non supera i 30 metri² e vi lavorano simultaneamente fino a circa 15 operatori. Non dispone della stanza da usare come "spogliatoio" e non è caratterizzato da pressione positiva.

I principali strumenti presenti sono:

- *Bagnetto termostato*: Consiste in una vaschetta, riempita con acqua distillata con aggiunta di sostanze antimuffa, tenuta a temperatura costante di 37°C mediante un riscaldatore posto all'interno di essa. Nel bagnetto vengono posti i terreni, i sieri e tutte le sostanze che andranno in contatto con le cellule. I bagnetti vengono svuotati e puliti in maniera approfondita ogni 4-6 mesi.
- *Bioreattore*: Il bioreattore è un dispositivo che permette la creazione e propagazione tridimensionale dei tessuti in coltivazione cellulare. Il

principio di funzionamento teorico del bioreattore è quello di simulare l'assenza di gravità. Questo dispositivo è un brevetto NASA che per prima ha effettuato esperimenti atti a osservare la particolare crescita cellulare nello spazio e ha cercato di simularne le stesse condizioni nell'atmosfera terrestre.

- Cappe a flusso laminare: sono utilizzate in ambito biologico per la protezione dell'operatore e dell'ambiente circostante da parte di agenti biologici (generalmente microrganismi patogeni); inoltre, la cappa a flusso laminare elimina la possibilità di contaminazioni crociate, consentendo un lavoro in condizioni di sterilità. La sterilizzazione dell'aria all'interno della cappa viene realizzata forzandone il passaggio attraverso filtri HEPA (*High Efficiency Particulate Air*): tali filtri sono realizzati in micro fibra di vetro che garantiscono aria "pura" al 100% (come prefissato dalla normativa della Legge 626).
- Centrifughe: sono apparecchiature impiegate per accelerare la separazione tra corpi aventi differente densità mediante l'uso dell'accelerazione centrifuga.
- Contenitori: generalmente sono di plastica (polistirene) monouso e sono vendute in confezioni sigillate e sterilizzate. Fondamentalmente si hanno due tipi di contenitori: le fiasche, sorta di bottiglie con tappo a vite, che possono avere il "corpo" cilindrico nel caso dei cosiddetti "roller" (cioè sistemi in cui la bottiglia viene mantenuta in continua rotazione, e serve per colture in grande scala), oppure a forma di parallelepipedo con una base stretta a 2 facce laterali ampie; le piastre, una sorta di scatola con una base e un coperchio. Le piastre Petri sono di solito rotonde, le piastre multipozzetto (meglio conosciute come *multiwell*) sono piastre quadrate suddivise in diversi pozzetti (da 6 a 96) a volume variabile a seconda del numero di pozzetti.
- Frigoriferi e congelatori: servono per conservare il materiale come terreni, sieri, reagenti e anche colture cellulari. Le temperature di conservazione variano dai 4°C ai -20°C, fino ad arrivare addirittura a -80°C (per la conservazione di cellule per tempi relativamente brevi, circa 6 mesi) o a -196°C (temperatura dell'azoto liquido usato per la conservazione a tempo indefinito delle cellule).

- *Incubatori*: utilizzati per mantenere il progressivo sviluppo delle colture microbiologiche regolando fattori di crescita sostenibile come temperatura, umidità, ventilazione e i livelli di CO₂.
- *Microscopi*: necessario sia quello diritto per contare la cellule e per l'osservazione di vetrini, sia quello rovesciato (o invertito) per osservare le cellule in cultura.
- *Pipette automatiche e pipettatori*: servono per prelevare volumi definiti di liquido. In laboratorio è sempre bene avere un set completo di pipette, che coprano un *range* di volumi almeno compreso tra 1 e 1000 μL .

Entrando più nel dettaglio, il microscopio usato per l'acquisizione delle immagini è uno ZEISS AXIOVERT 200 (Figura 5.1). È un microscopio invertito (fonte luminosa sopra il portaoggetti e obiettivi collocati sotto) utilizzato tipicamente per l'analisi di colture cellulari. Tale meccanismo permette di poter osservare le colture cellulari senza doverle mettere in un vetrino, indispensabile per i vari processi di routine di un laboratorio biologico (determinazione confluenza, stato delle cellule, ecc.) senza rendere inutilizzabili successivamente i campioni esaminati.



(a) Lato sinistro microscopio Axiovert 200 (b) Lato destro microscopio Axiovert 200

Figura 5.1: Microscopio Zeiss Axiovert 200

Tale microscopio può essere usato sia in brightfield sia in contrasto di fase.

Gli obiettivi disponibili sono:

1. ACHROPLAN 5x (abbinato all'anello di fase ph0)
2. ACHROPLAN 10x (abbinato all'anello di fase ph1)
3. LD APLAN 20x (abbinato all'anello di fase ph1)
4. ACHROPLAN 40x (abbinato all'anello di fase ph2)

I filtri disponibili sono:

1. B (invia al campione la luce nella lunghezza d'onda di eccitazione nel range del blu (488 nm) e lascia arrivare al rivelatore la luce emessa a lunghezza d'onda nel range del verde (530 nm).
2. G (invia al campione la luce nella lunghezza d'onda di eccitazione nel range del verde (530 nm) e lascia arrivare al rivelatore la luce emessa a lunghezza d'onda nel range del rosso (620 nm).
3. UV (invia al campione la luce nella lunghezza d'onda di eccitazione nel range dell'ultravioletto (400 nm) e lascia arrivare al rivelatore la luce emessa a lunghezza d'onda nel range del blu (488 nm).

Il condensatore ha una numerical aperture (NA) da 0.55 con 6 posizioni e gli anelli disponibili sono:

1. Ph0
2. ph1
3. ph2
4. H
5. DIC
6. DIC

Questo microscopio permette l'osservazione delle cellule anche in contrasto di fase, sia positivo che negativo:

- Nel contrasto di fase positivo se l'indice di rifrazione del campione è maggiore dell'indice di rifrazione del bianco, nell'immagine finale il soggetto si vede scuro.

- Nel contrasto di fase negativo se l'indice di rifrazione del campione è maggiore dell'indice di rifrazione del bianco, nell'immagine finale il soggetto si vede chiaro.

Come “bianco” in analisi cellulare tipicamente viene considerato un pozzetto contenente solo terreno di coltura (senza quindi cellule seminate). Nel laboratorio di Colture Cellulari dell'IRST questo microscopio è abbinato ad una fotocamera commerciale Zeiss AxioCam MRc (Figura 5.2).



Figura 5.2: Fotografia della fotocamera Zeiss AxioCam MRc.

La sigla MR indica i modelli *top* delle camere Zeiss, mentre il pedice “c” indica che la fotocamera è a colori (se fosse stata in bianco e nero, indicata nel caso venisse utilizzata unicamente per fotografie in fluorescenza, avrebbe avuto il pedice “m”). Questa fotocamera è stata acquistata dall'IRST circa cinque anni fa ed ora le performance di questa fotocamera, per quanto fossero ottimali qualche anno fa, risultano superate: con una capacità risolutiva di 1,4 *Megapixel* non si ottengono immagini di alta qualità. La maggiore limitazione di questa camera è l'impossibilità di acquisire video. Le caratteristiche tecniche fornite direttamente dalla casa costruttrice sono le seguenti (Figura 5.3, Figura 5.4).

Il *software* di acquisizione di immagini abbinato a questa camera, Axio-Vision AC, viene dato in dotazione insieme alla fotocamera dalla Zeiss stessa. Il *software* in questione risulta quasi immutato salvo piccole inezie da più di 7 anni, e per motivi aziendali/commerciali non permette l'acquisizione di video. Tramite il microscopio usato si riescono a fotografare aree di

Technical Data AxioCam MRc

Sensor	Sony ICX 285, progressive readout, with RGB filter mask			Signal amplification	Analog: 2x, digital 32x
CCD basic resolution	1388 x 1040 = 1,4 megapixels			Digitization	12 bit
Pixel size	6.45 µm (H) x 6.45 µm (V)			CCD cooling	One stage Peltier cooling
Sensor size	Chip area 8.9 mm x 6.7 mm, equivalent 2/3"			Interface	FireWire 1394a (400 megabits/s)
Spectral range	Approx. 400 nm-700 nm, BG 40 IR protection glass			Range of integration time	1 ms up to 60 s
Dynamic range	Typical > 1 : 2200 (> 66.8 dB)			Signal output connectors	2 x TTL-Out, exposure time and readout time (i.e. for driving external electric shutters), 1 x Trigger-In to start an acquisition
Full Well	Typical 17 Ke			Optical interface	C-Mount
Readout noise	Typical < 7.7 e			Housing	Blue anodized aluminum, with cooling fins, 1/4" connection for tripod mount, 11 cm x 8 cm x 4.5 cm / 370 g
Dark current	Typical 0.7 e/pixels/s, dark current compensation for maximum low light performance			Operating system	Microsoft® Windows 2000 Professional Microsoft® Windows XP Professional
Readout speed	24.57 MHz pixel clock			Registration	CE, CLIL
Live image frame rates	H	x	V	Mode / Binning	Max. frame rate*
	1388	x	1040	slow / 1	13 images/s
	460	x	344	middle / 3	26 images/s
	276	x	208	fast / 5	38 images/s
Resolution and frame rates for time lapse images in	H	x	V	Binning	Max. frame rate*
	1388	x	1040	1 x 1, RGB / SW	14 images/s
AxioVision module Fast	692	x	520	2 x 2, 5W	26 images/s
Acquisition (High Speed Color Mode**)	460	x	344	3 x 3, RGB	35 images/s
	344	x	260	4 x 4, 5W	42 images/s
	276	x	208	5 x 5, RGB	48 images/s
Max. file size per image	Approx. 8.6 MB at 1388 x 1040 at 3 x 12 bit (36 bit color depth)				
High-speed operation modes for AxioVision module Fast Acquisition	<ul style="list-style-type: none"> • Five preloadable exposure time parameters in camera head for high-speed multichannel acquisition*** • Continuous mode for fast triggered acquisition • Overlapping exposure and readout of the sensor in fast time lapse images**** 				
Color interpolation	High Speed Color Mode or High Quality Color Mode selectable				
Hard disk recording	Inline recording of image data directly to hard disk, at all speeds with AxioVision module Fast Acquisition				
Readout of subframes (ROI)	Freely selectable				

Above frame rates are supported by the camera electronics. Computer hardware, operating system and application software may decrease the frame rates. Selecting a part of the sensor area can increase the frame rate. All specifications are subject to change without notice.

* Frame rates depend on exposure time and readout mode.
 ** Image rates when recording onto hard drive in High Speed Color Mode.
 *** In Continuous Mode the maximal exposure time is 819 ms per channel.
 **** In basic resolution mode the sensor readout time is 69 ms. Below this value, the frame rate is only determined by readout time. Above this value, the frame rate is determined by exposure time, only. With activated binning mode, the readout time is shorter, respectively.

Figura 5.3: Caratteristiche tecniche fotocamera Zeiss MRc.

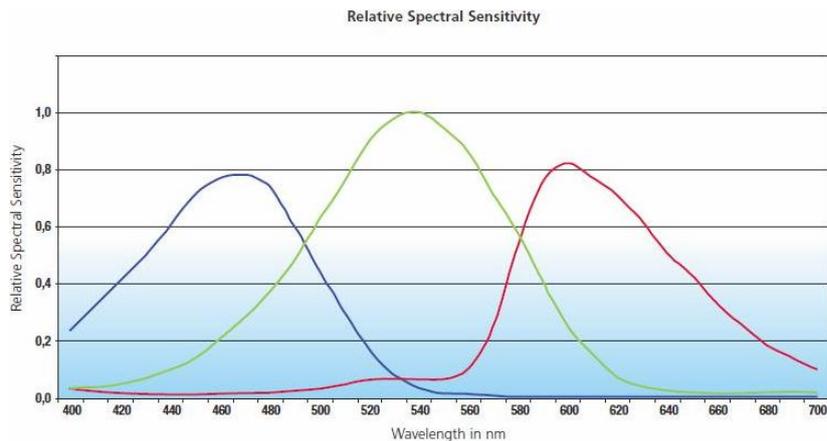


Figura 5.4: Immagine della sensibilità dello spettro RGB.

dimensioni varie a seconda dell'ingrandimento utilizzato, in particolare con i seguenti ingrandimenti queste sono le aree che si riescono a fotografare in un singolo frame di acquisizione:

- obiettivo 5x, area fotografata in un singolo frame: $2,2016 \text{ mm}^2$,
- obiettivo 10x, area fotografata in un singolo frame: $0,5545 \text{ mm}^2$,
- obiettivo 20x, area fotografata in un singolo frame: $0,1377 \text{ mm}^2$,
- obiettivo 40x, area fotografata in un singolo frame: $0,0367 \text{ mm}^2$.

5.2 Broncofere

Il gruppo di ricerca del laboratorio di Farmacologia Preclinica dell'IRST, capeggiato dai miei referenti il Dott. Wainer Zoli e la Dott.ssa Anna Tesei, è stato il primo al mondo ad isolare cellule staminali polmonari normali e tumorali e creare broncofere in vitro (Figura 5.5) che, come già detto, sono degli aggregati multicellulari a forma sferica (detti anche sferoidi) di cellule provenienti dai bronchi alveolari [44]. In particolare, all'IRST sono in fase di sperimentazione due diverse procedure per raggiungere il medesimo obiettivo, quale la creazione di sferoidi di cellule polmonari. Una prevede l'utilizzo di un terreno selettivo che permette la sopravvivenza delle sole cellule staminali, le quali riescono ad aggregarsi formando delle sferule. L'altra si basa sull'ausilio di un bioreattore (brevetto NASA) mediante il quale si effettua un processo meccanico che simula l'assenza di gravità favorendo il formarsi di aggregati multicellulari. L'essere riusciti ad isolare e ricreare degli sferoidi *in vitro* è un risultato molto importante in quanto risultano essere un ottimo modello di tessuto reale. In particolare, gli sferoidi ottenuti da linee stabilizzate di cellule tumorali sono un ottimo modello di tumore solido su cui testare farmaci e sperimentare terapie di cura, quali ad esempio i parametri di esposizione in radioterapia.

In particolare, utilizzando una coltura di sferoidi è stata acquisita una sequenza di immagini a diversi piani focali utilizzata come *data set* per testare i vari algoritmi di fusione descritti nel Capitolo 2 e nel Capitolo 3. Il *data set* è stato creato ritagliando dalle immagini originali (Figura 5.6) acquisite tramite microscopio in *brightfield*, una ROI in cui è contenuto lo sferoide di maggiori dimensioni (Figura 5.7).

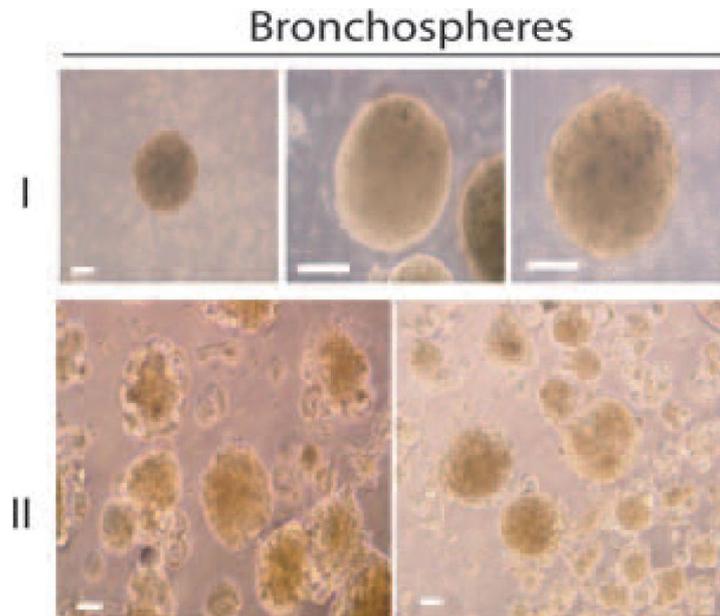


Figura 5.5: Broncofere generate da tessuto polmonare sano. (I) Immagine al microscopio invertito a contrasto di fase di broncofere primarie ottenute da tessuto polmonare; (II) broncofere secondarie derivate da broncofere primarie coltivate in terreno privo di siero. La barra di scala è $100 \mu m$. Cortesemente da [44].

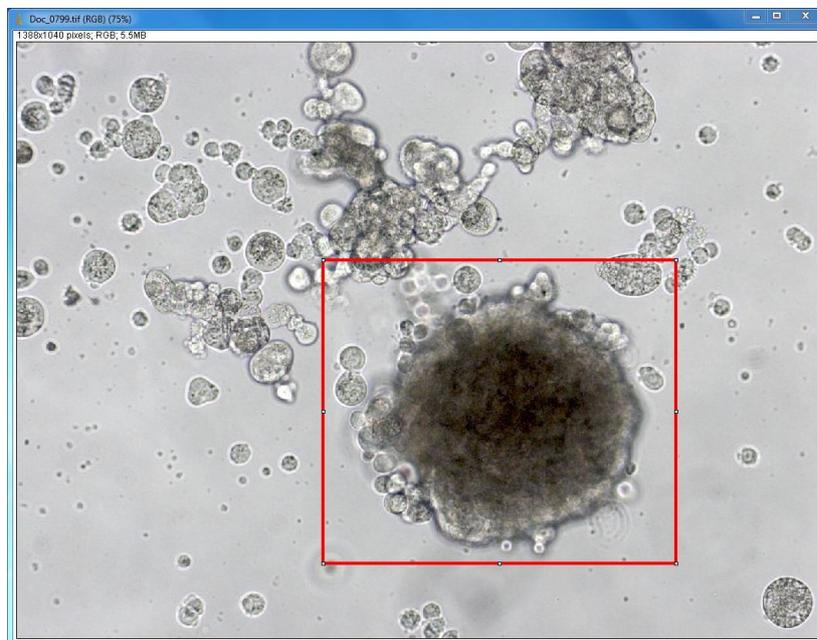


Figura 5.6: Esempio di immagine di sferoide acquisita con microscopio in *brighfield*. L'area dell'immagine all'interno del quadrato rosso sarà la ROI che verrà ritagliata da ogni immagine dello *stack*.

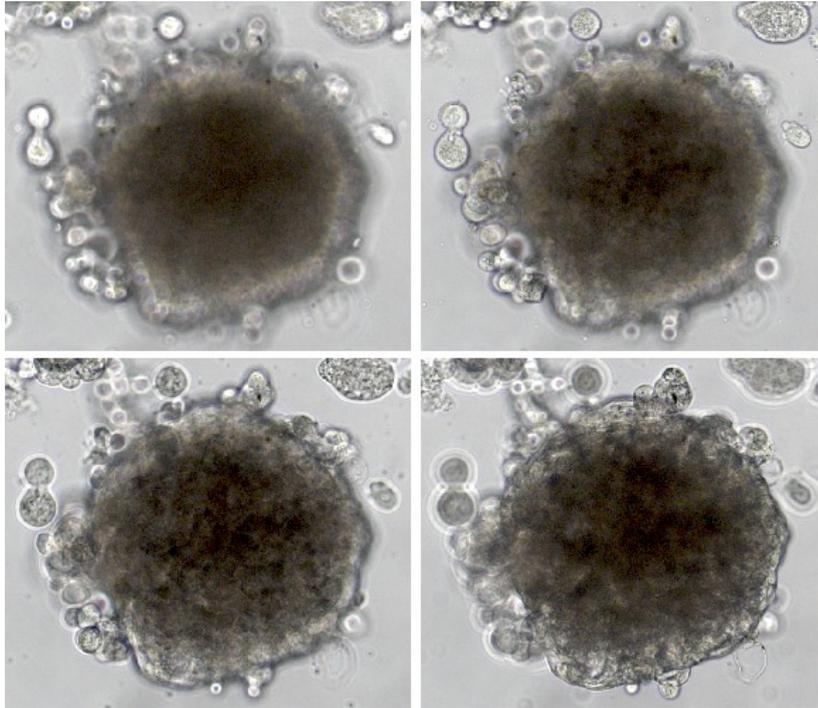


Figura 5.7: Ritagli della ROI di diverse immagini dello *stack* prese a differenti piani focali.

5.3 *Data set* artificiali

Per un'analisi approfondita delle performance ottenibili dall'algoritmo per l'estensione della profondità di campo da me proposto nel Capitolo 3, rispetto a quelle ottenibili sfruttando alcuni metodi allo stato dell'arte per l'*image fusion*, sono stati utilizzati diversi data set di calibrazione costituiti da immagini sintetiche (in inglese *phantom data*) e diversi data set di immagini reali. L'utilizzo dei *phantom data* ha permesso di utilizzare tutte le metriche per la valutazione della qualità delle immagini finali come l'MSE, l'SNR e l'UQI che necessitano della *ground truth* per essere calcolate. I *phantom data* (creati con otto immagini ciascuno) sono stati costruiti a partire da cinque immagini (Figura 5.8), molto diverse nei contenuti, del database delle Brodatz *texture* (Brodatz, 1966, <http://www.ux.uis.no/~tranden/brodatz.html>). In particolare sono state utilizzate le immagini denominate D13, D22, D23, D61, D112. D13 e D61 potrebbero assomigliare a dei tessuti ossei, D22 ad una biopsia polmonare, D23 e D112 a delle culture cellulari.

Per costruire gli *stack* delle immagini sintetiche è stata sfruttata la funzione specifica, accessibile dalla sua interfaccia grafica del *plug-in* ImageJ del software MBM e (Figura 2.3), già utilizzata e ben spiegata in [2]. La

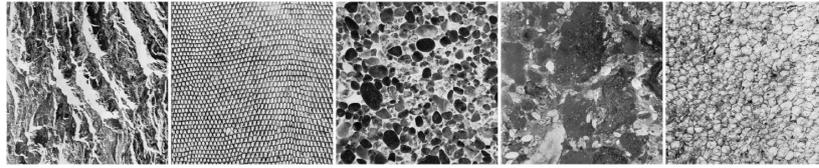


Figura 5.8: Immagini in scala di grigi dal database delle Brodatz *texture*. Da sinistra a destra: D13, D22, D23, D61, D112.

funzione proietta una *texture* (nel nostro caso le 5 Brodatz *texture*) su una superficie (Figura 5.9 (a)). Per creare le *slides* il volume è poi discretizzato attraverso una interpolazione lineare lungo la direzione z . In seguito, per simulare il sistema ottico, il volume è convoluto con una PSF gaussiana con larghezza crescente all'aumentare della distanza focale (Figura 5.9 (b)). La somma dei coefficienti dei filtri per la PSF gaussiana ad ogni distanza focale è normalizzata, ciò corrisponde al caso dell'illuminazione di Köhler.

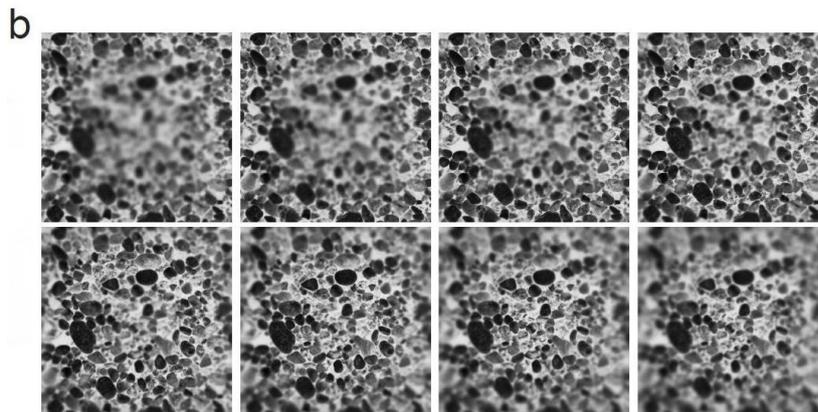


Figura 5.9: (a) Brodatz *texture* D23 proiettata su una superficie a forma di cupola; (b) Sequenza di otto immagini ottenuta dall'immagine (a) attraverso la convoluzione con la PSF gaussiana.

5.4 *Data set* reali

Sono stati utilizzati 5 *data set* di immagini reali per testare i diversi algoritmi. Uno *stack* di immagini reali è stato creato, come già detto, utilizzando una sequenza di immagini di sferoide acquisita al laboratorio di Farmacologia Preclinica dell'IRST. Pur avendo a disposizione altre sequenze di sferoidi, non sono state usate perché si sono voluti testare gli algoritmi studiati su immagini diverse provenienti da differenti microscopi per garantire una maggior generalità del metodo proposto. Gli altri quattro *data set* utilizzati

sono stati gentilmente inviati dal responsabile dello sviluppo dei *software* del *Biomedical Imaging Group* (BIG) dell' *École Polytechnique Fédérale de Lausanne* (EPFL), il Dr. Daniel Sage (<http://bigwww.epfl.ch/sage/>). I quattro *stack* sono:

- Stack composto da 13 immagini a colori di una saldatura laser (Figura 5.10);
- Stack composto da 20 immagini a colori rappresentante le placche di Peyer dall'intestino di topo (Figura 5.11);
- Stack composto da 17 immagini a colori del processo unicinato del pancreas (Figura 5.12);
- Stack composto da 32 immagini in scala di grigi a 32-bit di un occhio di mosca (Figura 5.13).

In particolare, questi *data set* di immagini reali sono quelli che sono già stati utilizzati nelle seguenti pubblicazioni [17], [2], [45] al fine di testare metodi o settare parametri di alcuni algoritmi.

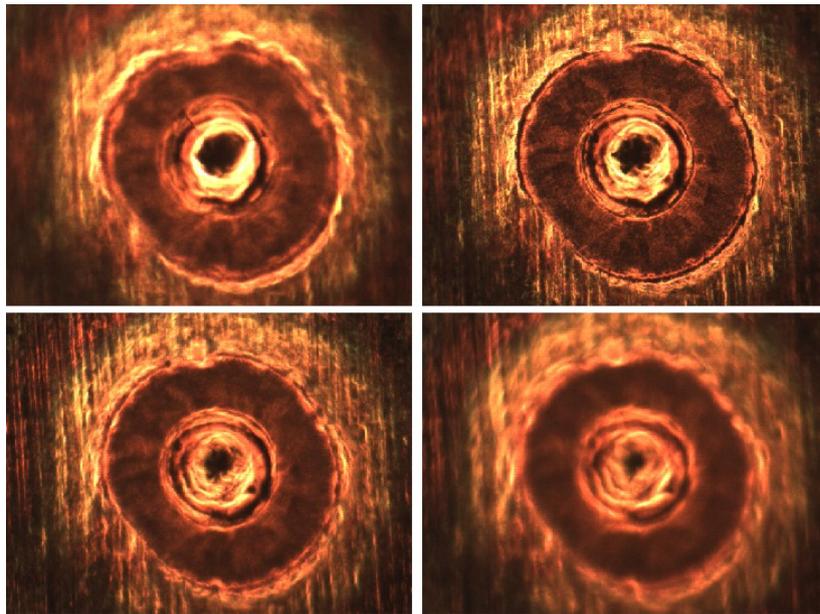


Figura 5.10: Alcune immagini prese dalla sequenza della saldatura laser.

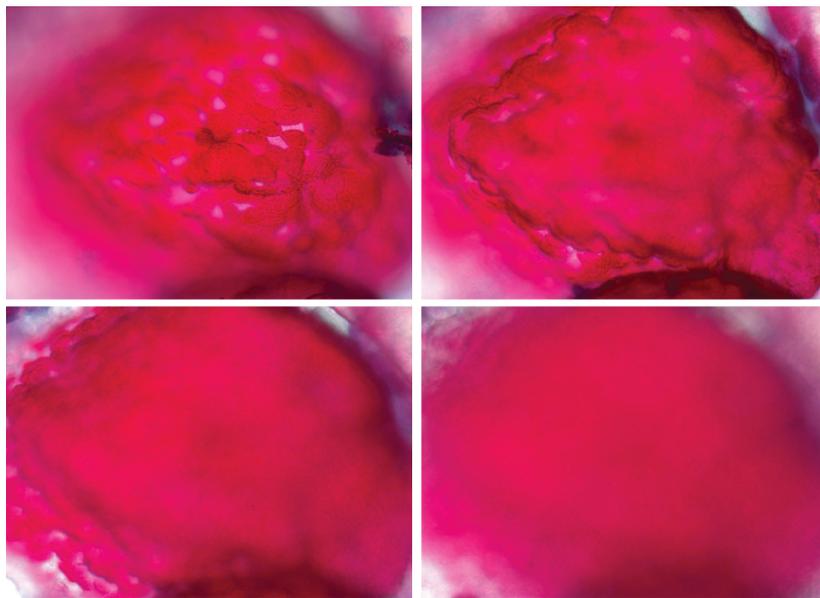


Figura 5.11: Alcune immagini prese dalla sequenza delle Peyer dall'intestino di topo.

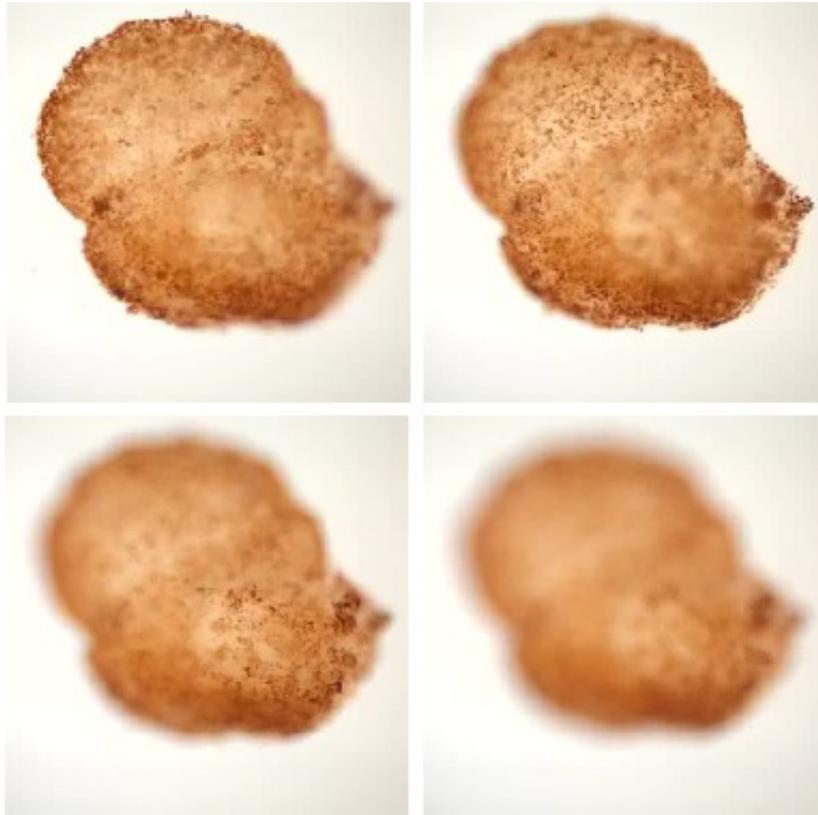


Figura 5.12: Alcune immagini prese dalla sequenza del processo uncinato del pancreas.

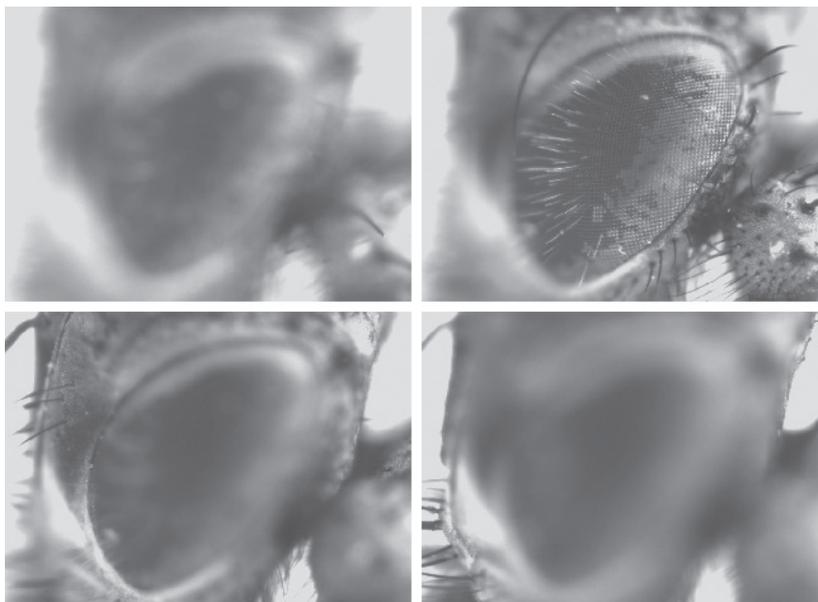


Figura 5.13: Alcune immagini prese dalla sequenza dell'occhio di mosca.

Capitolo 6

Risultati Sperimentali

In questo capitolo verranno presentati e discussi i risultati ottenuti dai vari test effettuati utilizzando i *data set* sintetici e reali mostrati nel Capitolo 5. In particolare, sfruttando gli *stack* di immagini sintetiche correlate da *ground truth* e le metriche MSE, SNR e UQI *standard*, verrà mostrato che il nuovo metodo proposto per estendere la profondità di fuoco, denominato AnPuLo, fornisce immagini composite finali migliori rispetto a quelle degli altri metodi. Inoltre, sempre utilizzando i *data set* di immagini sintetiche e le immagini composite finali ottenute sfruttando i vari metodi presentati nel Capitolo 2, verrà dimostrato che le metriche PM e UQIN, nonostante siano proposte come estensioni dell'UQI, non sono in grado di classificare i metodi testati allo stesso modo dell'UQI *standard*. Verrà quindi proposto e validato un nuovo approccio in grado di classificare i diversi metodi per l'estensione della profondità di fuoco alla stessa identica maniera dell'UQI *standard*, ma senza la necessità di richiedere la *ground truth*.

6.1 *Testbed* dei parametri dell'algoritmo

Nell'algoritmo proposto nel Capitolo 3, come già detto, si è deciso di separare lo stadio di rivelamento dei contorni dallo stadio di filtraggio sulle derivate in modo da poter avere un maggior controllo di entrambi i processi. In particolare, dopo aver rilevato i contorni, su ogni immagine dello *stack* di *slice* viene applicato un filtro a media con *kernel* di dimensioni 15×15 . La dimensione del filtro è stata decisa dopo aver eseguito un test granulometrico sfruttando le immagini D22, D23 e D112. Dai test effettuati, la finestra 15×15 è risultata essere il compromesso ideale tra dimensione del filtro locale e massimizzazione dell'SNR (Figura 6.1).

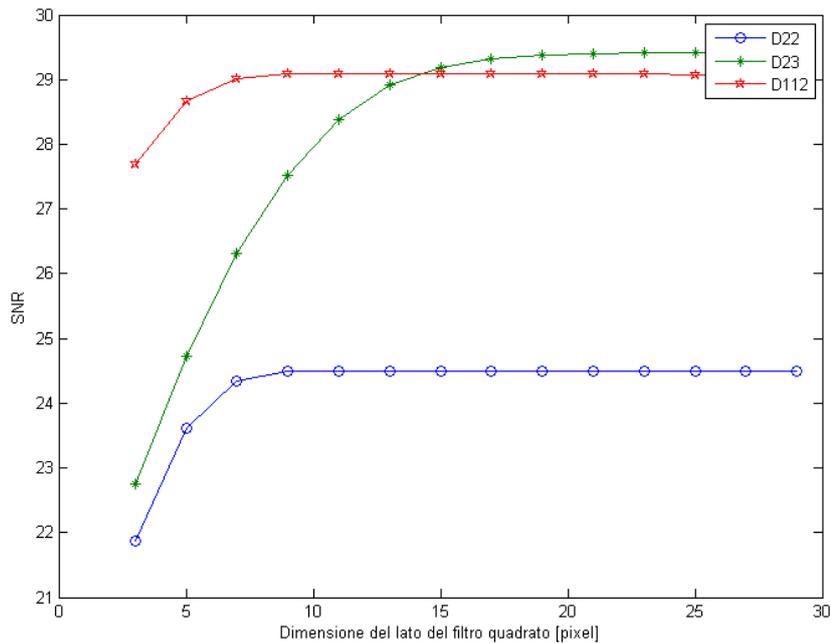


Figura 6.1: Grafico che mostra i risultati del test granulometrico.

6.2 Comparazione metodi utilizzando immagini sintetiche

Per confrontare le prestazioni dell’algoritmo per l’estensione della profondità di campo presentato nel Capitolo 3, con i metodi allo stato dell’arte descritti nel Capitolo 2, le immagini composite finali completamente a fuoco ottenute dall’elaborazione dei *data set* sintetici sono state comparate utilizzando le immagini di *ground truth* (cioè le stesse Brodatz *texture* usate per costruire gli *stack*) e le metriche MSE, SNR e UQI. In particolare, in Tabella 6.1 sono riportati i valori ottenuti rispettivamente per l’MSE, l’SNR e l’UQI. Per tutte le immagini e per tutte le metriche, i migliori risultati ottenuti (intesi come l’MSE più basso, l’SNR e l’UQI più alti) sono sempre quelli relativi al nuovo metodo proposto per l’estensione della profondità di fuoco. È opportuno notare che per ogni *data set* la classifica ottenuta ordinando i metodi dal migliore al peggiore, è la stessa per tutte le metriche e per tutti i *data set* (Tabella 6.2). Non è un dato di fatto ottenere la stessa classifica per tutte e tre queste metriche, ma quando questo si verifica [11], [5], [19] la classifica dei metodi ottenuta può essere considerata molto affidabile. Analizzando i dati, è apprezzabile che i risultati migliori sono sempre seguiti dai valori ottenuti dallo SF (che è anche il metodo più simile a quello da me proposto) e in generale si nota che tutti i metodi portano a risultati

abbastanza simili ad eccezione dello SI. In particolare, i risultati numerici ottenuti dallo SI sono decisamente i peggiori. Questo è anche confermato da un'analisi visiva (Figura 6.2). Generalmente, nelle applicazioni di *image fusion* un risultato visivo non buono è inteso come un'immagine finale contenente rumore evidente, dischi di Airy, o artefatti prodotti per esempio da diffrazione o *aliasing* (Figura 6.2(d)) [20].

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
MSE D13	77,97	57,17	291,44	53,38	51,03	58,12	68,69
MSE D22	143,50	100,23	415,05	97,53	95,31	106,43	137,62
MSE D23	32,58	24,85	244,39	23,43	21,86	25,47	29,88
MSE D61	23,14	18,97	153,26	16,94	15,87	19,09	21,45
MSE D112	54,29	42,35	281,93	39,02	37,63	43,84	51,29
SNR D13	24,68	26,03	18,95	26,32	26,52	25,96	25,23
SNR D22	22,71	24,27	18,10	24,39	24,49	24,01	22,89
SNR D23	27,46	28,63	18,71	28,89	29,19	28,53	27,83
SNR D61	28,11	28,98	19,90	29,47	29,75	28,95	28,44
SNR D112	27,50	28,58	20,35	28,94	29,09	28,43	27,75
UQI D13	0,9915	0,9938	0,9676	0,9943	0,9945	0,9937	0,9925
UQI D22	0,9837	0,9889	0,9548	0,9892	0,9895	0,9881	0,9843
UQI D23	0,9961	0,997	0,9691	0,9972	0,9974	0,9970	0,9964
UQI D61	0,9949	0,9958	0,9663	0,9963	0,9965	0,9958	0,9953
UQI D112	0,9885	0,9911	0,9433	0,9919	0,9922	0,9908	0,9891

Tabella 6.1: MSE, SNR e UQI calcolati usando le *ground truth* e le immagini composite finali ottenute per tutti gli algoritmi testati e per tutte 5 Brodatz *texture* utilizzate.

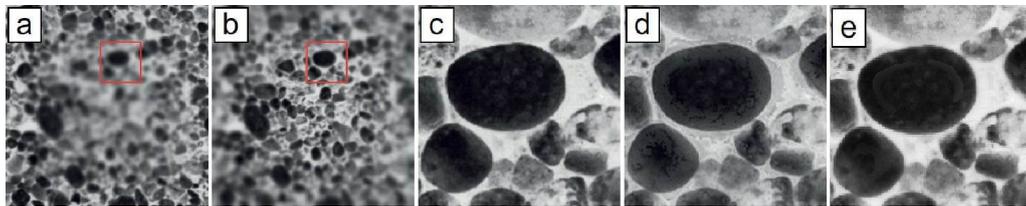


Figura 6.2: (a) e (b): due *slice* dello *stack* creato artificialmente usando l'immagine D23. (c): dettaglio ingrandito (posizione evidenziata nelle immagini (a) e (b) dal quadrato rosso) estratto dalla *ground truth* dell'immagine D23. (d) e (e): stesso dettaglio di (c), ma provenienti dalle immagini ottenute rispettivamente con i metodi SI e AnPuLo.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
MSE D13	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
MSE D22	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
MSE D23	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
MSE D61	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
MSE D112	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
SNR D13	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
SNR D22	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
SNR D23	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
SNR D61	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
SNR D112	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
UQI D13	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
UQI D22	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
UQI D23	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
UQI D61	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th
UQI D112	6 th	3 rd	7 th	2 nd	1 st	4 th	5 th

Tabella 6.2: Classifiche ottenute partendo dai valori in Tabella 6.1

6.3 Validazione delle metriche estensione dell'UQI

Come già detto nel Capito 4 , molte metriche sono state proposte in letteratura per valutare numericamente la qualità dell'*output* dei metodi di fusione usando *data set* di immagini reali, ovviamente non correlati da *ground truth*. In particolare, le metriche PM e UQIN sono presentate come estensione dell'UQI *standard* e sono ampiamente utilizzate in letteratura. Purtroppo nessuna analisi esaustiva è stata eseguita per dimostrare che queste metriche sono in grado di valutare la qualità delle immagini allo stesso modo dell'UQI, ma senza la necessità di disporre della *ground truth*. Per studiare in modo approfondito questa correlazione si è deciso di utilizzare i *data set* di immagini sintetiche costruiti (che sono quindi correlati da *ground truth*) e di valutare come tutte le tre versioni proposte per l'UQIN classificano i vari metodi testati ordinandoli dal migliore al peggiore. Se queste metriche fossero in grado realmente di valutare come l'UQI *standard*, si otterrebbe sempre la stessa identica classifica. In Tabella 6.3 sono riportati i risultati associati alle tre formulazione dell'UQIN usando una finestra mobile di 9×9 *pixel* e coefficiente α pari ad 1 per la terza formula dell'UQIN (Equazione (4.8)). La Tabella 6.4 riporta le classifiche ottenute analizzando i dati della tabella precedente (Tabella 6.3). Come si può notare, non è mai stata ottenuta la

stessa classifica raggiunta usando l'UQI *standard* (Tabella 6.2). Ciò dimostra che queste metriche non sono in grado di valutare allo stesso modo dell'UQI *standard*. Inoltre, questa conclusione è enfatizzata dal fatto che usando nelle Equazioni (4.4),(4.7) e (4.8) la *ground truth* come immagine f (al posto quindi dell'immagine finale a fuoco), i risultati ottenuti non sono i migliori, cioè non sono stati ottenuti valori più alti utilizzando la *ground truth* invece che le immagini a fuoco ottenute dai vari metodi. Questo logicamente evidenzia i limiti di queste metriche proposte in letteratura (Tabella 6.5). Tuttavia, vale la pena notare che anche con queste metriche, come con l'MSE, l'SNR e l'UQI *standard*, i risultati peggiori sono sempre ottenuti utilizzando il metodo SI.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
UQIN D13	0,8035	0,7878	0,7693	0,7761	0,7849	0,7897	0,7980
UQIN _W D13	0,8621	0,8504	0,8286	0,8504	0,8503	0,8512	0,8577
UQIN _E D13	0,8467	0,8368	0,7889	0,8379	0,8379	0,8377	0,8415
UQIN D22	0,8899	0,8797	0,8502	0,8801	0,8799	0,8804	0,8869
UQIN _W D22	0,8950	0,8856	0,8627	0,8858	0,8857	0,8862	0,8925
UQIN _E D22	0,8572	0,8493	0,8048	0,8499	0,8499	0,8495	0,8527
UQIN D23	0,8178	0,8019	0,7736	0,7941	0,7947	0,803	0,8123
UQIN _W D23	0,8828	0,8734	0,8499	0,8729	0,8725	0,8741	0,8801
UQIN _E D23	0,8402	0,8279	0,7546	0,8288	0,8286	0,8289	0,8343
UQIN D61	0,8073	0,7940	0,7732	0,7945	0,7953	0,7962	0,8036
UQIN _W D61	0,8554	0,8445	0,8206	0,8451	0,8449	0,8459	0,8520
UQIN _E D61	0,8334	0,8225	0,7758	0,8242	0,8244	0,8237	0,8279
UQIN D112	0,8283	0,8157	0,7909	0,8165	0,8167	0,8174	0,8250
UQIN _W D112	0,8600	0,8497	0,8274	0,8502	0,8500	0,8507	0,8572
UQIN _E D112	0,8496	0,8412	0,7973	0,8427	0,8427	0,8418	0,8452

Tabella 6.3: Valori ottenuti per le tre differenti versioni proposte per la metrica UQIN sfruttando tutti i metodi di ricostruzione presentati e le 5 Brodatz *texture*.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
UQIN D13	1 st	4 th	7 th	6 th	5 th	3 rd	2 nd
UQIN _W D13	1 st	5 th	7 th	4 th	6 th	3 rd	2 nd
UQIN _E D13	1 st	6 th	7 th	3 rd	4 th	5 th	2 nd
UQIN D22	1 st	6 th	7 th	4 th	5 th	3 rd	2 nd
UQIN _W D22	1 st	6 th	7 th	4 th	5 th	3 rd	2 nd
UQIN _E D22	1 st	6 th	7 th	3 rd	4 th	5 th	2 nd
UQIN D23	1 st	4 th	7 th	6 th	5 th	3 rd	2 nd
UQIN _W D23	1 st	4 th	7 th	5 th	6 th	3 rd	2 nd
UQIN _E D23	1 st	6 th	7 th	4 th	5 th	3 rd	2 nd
UQIN D61	1 st	6 th	7 th	5 th	4 th	3 rd	2 nd
UQIN _W D61	1 st	6 th	7 th	4 th	5 th	3 rd	2 nd
UQIN _E D61	1 st	6 th	7 th	4 th	3 rd	5 th	2 nd
UQIN D112	1 st	6 th	7 th	5 th	4 th	3 rd	2 nd
UQIN _W D112	1 st	6 th	7 th	4 th	5 th	3 rd	2 nd
UQIN _E D112	1 st	6 th	7 th	4 th	3 rd	5 th	2 nd

Tabella 6.4: Classifiche relative alla tabella precedente (Tabella 6.3)

6.4 Nuovo approccio di valutazione

Per superare la mancanza di una metrica utilizzabile con data set di immagini reali, per le quali tipicamente non è presente la *ground truth*, e in grado di classificare i metodi come farebbe l'UQI *standard*, abbiamo proposto e validato un nuovo approccio basato sulla definizione di base dell'UQI. L'UQI, come dichiarato dagli autori, è in grado di valutare la somiglianza di due immagini allo stesso modo della percezione visiva umana, nonostante nessun modello ispirato al sistema visivo umano sia in esso realizzato. Idealmente, un'immagine completamente a fuoco è teoricamente ben dettagliata e dovrebbe essere molto differente dalle *slice* sfuocate che compongono lo *stack*, in cui tipicamente solo una porzione di ciascuna immagine è a fuoco. Infatti, come detto nel Paragrafo 6.2, nelle applicazioni di *image fusion* è facile individuare visivamente quali metodi producono risultati visivamente non buoni. Partendo da queste due considerazioni, abbiamo validato un nuovo approccio in grado di classificare i vari metodi per l'estensione della profondità di fuoco come farebbe l'UQI *standard*, ma senza la necessità della *ground truth*. In primo luogo tutti i metodi che visivamente portano a risultati non buoni sono scartati attraverso un'analisi comparativa. In seguito, l'*Average UQI* (AUQI) è calcolato tra l'immagine finale a fuoco f (ottenuta da ogni metodo di fusione testato) e le N *slice* s_n dello *stack* [22], [23]:

$$AQUI(s_1, \dots, s_n, f) = \frac{1}{N} \sum_{n=1}^N UQI(s_n, f) \quad (6.1)$$

	Ground Truth
UQIN D13	0,7760
UQIN _W D13	0,8115
UQIN _E D13	0,7376
UQIN D22	0,8377
UQIN _W D22	0,8504
UQIN _E D22	0,7908
UQIN D23	0,7597
UQIN _W D23	0,8432
UQIN _E D23	0,7740
UQIN D61	0,7413
UQIN _W D61	0,8062
UQIN _E D61	0,7624
UQIN D112	0,8088
UQIN _W D112	0,7648
UQIN _E D112	0,7782

Tabella 6.5: Valori dell'UQIN utilizzando le immagini degli *stack* e la *ground truth* al posto dell'immagine *f*.

Teoricamente, la migliore immagine a fuoco ottenuta dai diversi metodi dovrebbe essere quella che maggiormente si discosta dalle *slice* sfuocate che compongono lo *stack*. Seguendo questa ipotesi, si è giudicato come metodo migliore quello che ha ottenuto il valore di AUQI più basso, e così via fino al metodo peggiore (tra quelli non scartati) inteso come quello con il valore di AQU I più alto, cioè quel metodo con l'immagine finale a fuoco più vicina alle *slice* sfuocate dello *stack*. È opportuno notare che l'AUQI, come l'UQI *standard*, è proposto come un indice globale senza alcuna funzione di peso implementata, mentre tutte le versioni dell'UQIN sono definite come un indice locale pesato. Questo potrebbe svolgere un ruolo importante nella classificazione delle immagini. Nella Tabella 6.6 sono riportati i valori dell'AQUI ottenuti sui cinque *data set* sintetici testati, scartando solo il metodo SI che ha sempre ottenuto immagini finali visivamente non buone. Seguendo l'approccio appena proposto, la classifica dei metodi così ottenuti è esattamente uguale a quella ottenuta usando MSE, SNR e UQI *standard*. Inoltre, usando la *ground truth* come immagine finale a fuoco (immagine *f* nell'Equazione (6.1)), e calcolando l'AUQI, i valori ottenuti sono sempre più bassi di quelli riportati nella Tabella 6.6, e questo è un'altra prova dell'affidabilità dell'approccio proposto (Tabella 6.7). È importante sottolineare che un passo fondamentale e cruciale di questo approccio è quello di scartare

i metodi che producono risultati visivi non buoni. Se non si scartassero questi metodi si potrebbero avere errori nella stima della classificazione: un'immagine finale composita contenente del rumore evidente, dischi di Airy, o artefatti potrebbe essere molto differente dalle immagini sfuocate che compongono lo *stack*, e calcolando l'AUQI tra questo tipo di immagine e le *slice* dello *stack* si potrebbe ottenere un valore molto molto basso ed il metodo associato potrebbe erroneamente essere quindi classificato come metodo migliore. Analizzando i dati nella Tabella 6.6, è opportuno notare che, come accade utilizzando MSE, SNR e UQI standard, per tutti i *data set* i risultati migliori (AUQI più basso) sono ottenuti dall'algoritmo AnPuLo, e sono sempre molto simili a quelli dello SF (in seconda posizione per tutti i *data set*). Nella Tabella 6.8 sono riportate le classifiche relative ai dati della Tabella 6.6.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
AUQI D13	0,8917	0,8831	0,8715	0,8823	0,8822	0,8842	0,8897
AUQI D22	0,6688	0,6543	0,6503	0,6541	0,6538	0,6560	0,6674
AUQI D23	0,9274	0,923	0,9157	0,9223	0,9220	0,9236	0,9270
AUQI D61	0,9360	0,9315	0,9071	0,9308	0,9306	0,9325	0,9354
AUQI D112	0,8363	0,8258	0,7868	0,8253	0,8250	0,8278	0,8353

Tabella 6.6: Valori dell'AUQI ottenuti con i vari metodi testati e i cinque *data set* artificiali.

Groung Truth	
AUQI D13	0,8510
AUQI D22	0,6068
AUQI D23	0,9057
AUQI D61	0,9104
AUQI D112	0,7820

Tabella 6.7: Risultati dell'AUQI utilizzando come immagine *f* la *ground truth*.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
AUQI D13	6 th	3 rd	/	2 nd	1 st	4 th	5 th
AUQI D22	6 th	3 rd	/	2 nd	1 st	4 th	5 th
AUQI D23	6 th	3 rd	/	2 nd	1 st	4 th	5 th
AUQI D61	6 th	3 rd	/	2 nd	1 st	4 th	5 th
AUQI D112	6 th	3 rd	/	2 nd	1 st	4 th	5 th

Tabella 6.8: Classifica relativa ai valori dell’AUQI della Tabella 6.6. Il simbolo “/” indica che quel metodo è stato scartato perché ha prodotto risultati visivi non buoni.

6.5 Comparazione metodi utilizzando immagini reali

Tipicamente, per tutti i *data set* di immagini reali, acquisiti spostando lungo la direzione z il tavolino portapreparati del microscopio (in inglese *holder*), non è disponibile la *ground truth*, intesa come singola immagine completamente a fuoco. A causa di questa informazione mancante, la qualità dell’immagine in uscita dal metodo per l’estensione della profondità di campo non può essere valutata sfruttando MSE, SNR e UQI *standard* perché tutte queste metriche richiedono un’immagine di riferimento. Inoltre, come mostrato nel Paragrafo 6.3, le metriche PM e UQIN non sono in grado di valutare le immagini allo stesso modo dell’UQI *standard*. In contrasto con queste metriche, l’approccio proposto al Paragrafo 6.3, basato sul calcolo dell’AUQI preceduto da una fondamentale analisi visiva dei risultati, ha mostrato una classifica finale completamente uguale a quella dell’UQI *standard* per tutti i *data set* sintetici testati. Questo dimostra che questo tipo di approccio potrebbe essere usato per valutare i metodi per l’estensione della profondità di campo anche in casi reali, dove tipicamente la *ground truth* non è a disposizione. È importante notare che, diversamente dagli *stack* sintetici, usando *data set* di immagini reali acquisite con differenti sistemi di acquisizione, intesi come microscopio, obiettivo e macchina fotografica, e diversi passi di spostamento lungo l’asse z tra le *slice*, è possibile ottenere una classifica dei metodi *data set-dipendente*, cioè diversa per ogni *data set* analizzato. Inoltre, gli *stack* sintetici sono solo una semplice approssimazione dei casi reali, in cui generalmente la PSF è spazio-variante nelle tre dimensioni e dipende da numerosi parametri quali lo spettro della sorgente luminosa, la distanza tra il vetrino e la superficie dell’oggetto, e le zone fuori

fuoco non sono correttamente modellate da un semplice filtro gaussiano. Come conseguenza di quello appena descritto, l'utilizzo di *data set* reali può portare ad ottenere delle classifiche dei metodi differenti da quelle stimate sui *data set* sintetici. Dalla Figura 6.3 alla Figura 6.7 sono riportate le immagini composite finali ottenute utilizzando i sette metodi testati sui cinque *stack* di immagini reali. Da un'analisi visiva comparativa è possibile notare che anche in questo caso il metodo SI è quello che fornisce le immagini peggiori in *output*. Nella Tabella 6.9 e nella Tabella 6.10 sono riportati rispettivamente i valori dell'AUQI e le classifiche dei metodi, ottenute semplicemente scaricando le immagini relative allo SI. Come previsto, le classifiche dei metodi sono risultate *data set-dipendenti* e inoltre sono risultate diverse da quella ottenuta utilizzando gli *stack* sintetici. Tuttavia, le prime due posizioni in classifica sono di nuovo occupate da SF e AnPuLo per tutti i *data set* e i valori ottenuti usando questi due metodi sono molto simili, quasi identici. Per completezza, nella Tabella 6.11 e nella Tabella 6.12 sono riportati i valori e la classifiche ottenute usando tutte le tre versioni dell'UQIN.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
AUQI Saldatura Laser	0,9086	0,9062	/	0,8996	0,9021	0,9123	0,9096
AUQI Intestino Topo	0,8466	0,8542	/	0,8278	0,8286	0,8730	0,8393
AUQI Pancreas	0,9561	0,9557	/	0,9413	0,9410	0,9570	0,9495
AUQI Sferoide	0,9422	0,9306	/	0,9271	0,9278	0,9434	0,9413
AUQI Occhio Mosca	0,8692	0,8947	/	0,8524	0,8557	0,8899	0,8904

Tabella 6.9: Valori AUQI per i *data set* reali.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
AUQI Saldatura Laser	4 th	3 rd	/	1 st	2 nd	6 th	5 th
AUQI Intestino Topo	4 th	5 th	/	1 st	2 nd	6 th	3 rd
AUQI Pancreas	5 th	4 th	/	2 nd	1 st	6 th	3 rd
AUQI Sferoide	5 th	3 rd	/	1 st	2 nd	6 th	4 th
AUQI Occhio Mosca	3 rd	6 th	/	1 st	2 nd	4 th	5 th

Tabella 6.10: Classifiche AUQI per i *data set* reali.

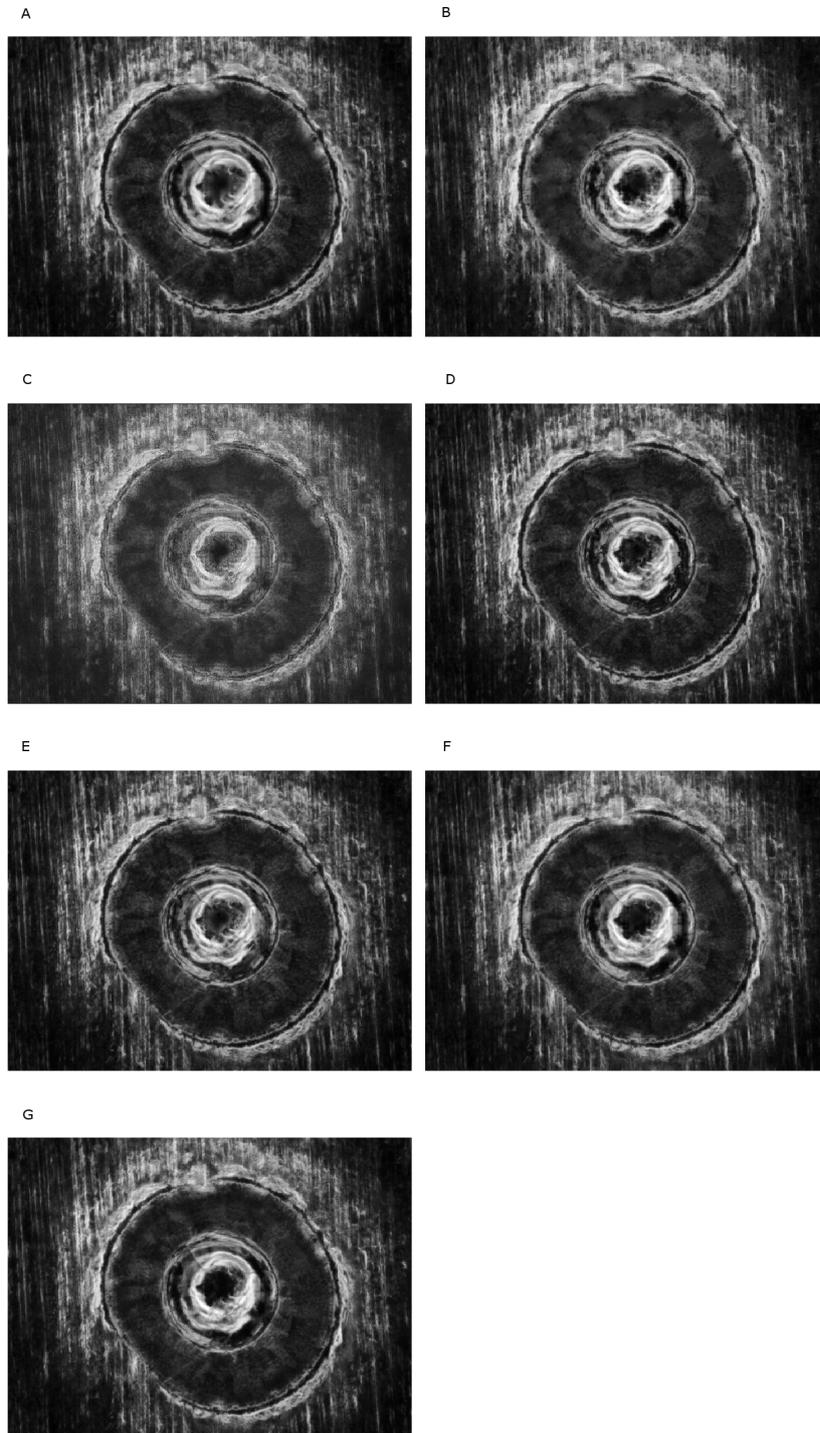


Figura 6.3: Immagini composite finali ottenute usando lo *stack* lo stack “saldatura laser”. Per ottenere le immagini sono stati usati i seguenti algoritmi: (A) MBM, (B) CWT, (C) SI, (D) SF, (E) AnPuLo, (F) DT-CWT, (G) CT.

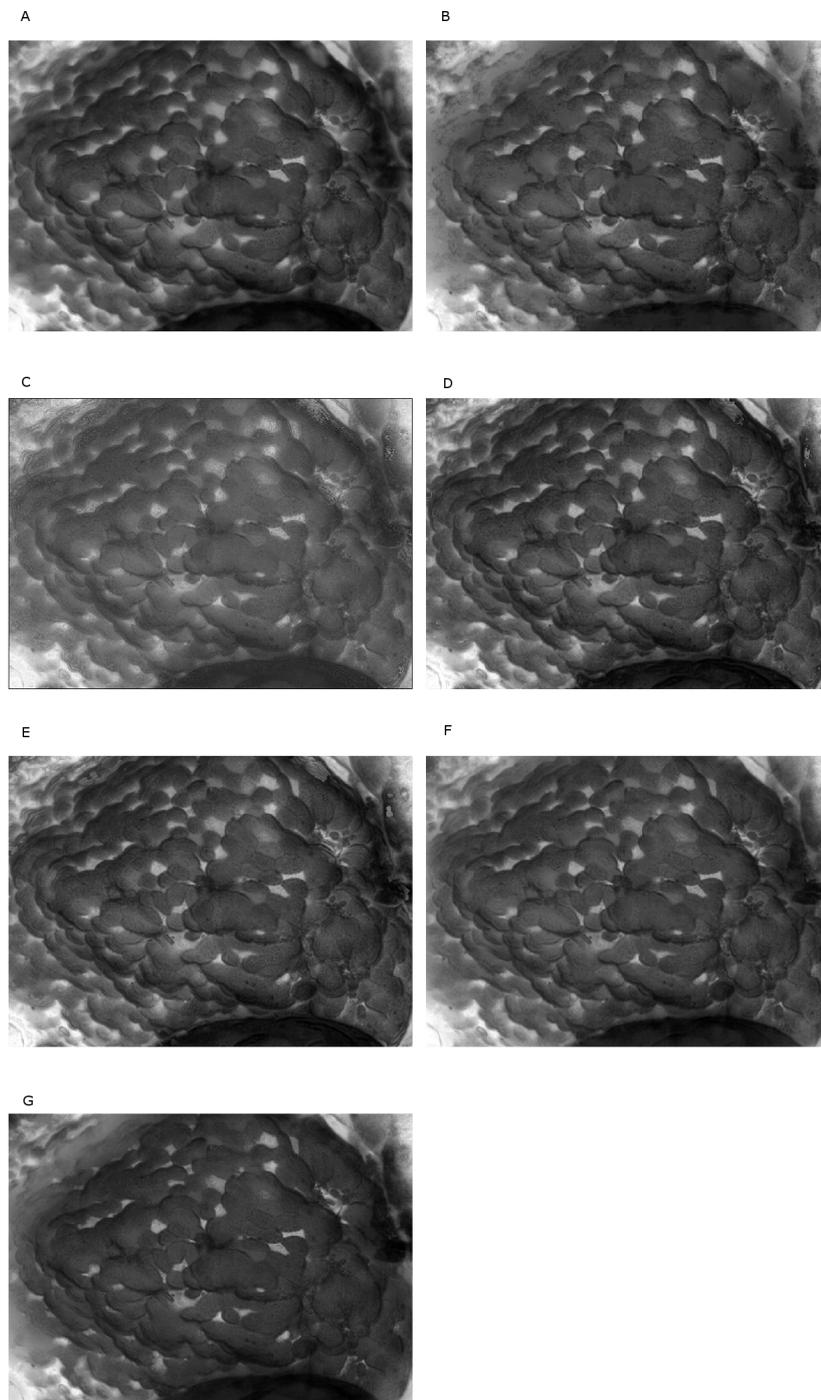


Figura 6.4: Immagini composite finali ottenute usando lo *stack* “placche di Peyer dell’intestino di topo”. Per ottenere le immagini sono stati usati i seguenti algoritmi: (A) MBM, (B) CWT, (C) SI, (D) SF, (E) AnPuLo, (F) DT-CWT, (G) CT.

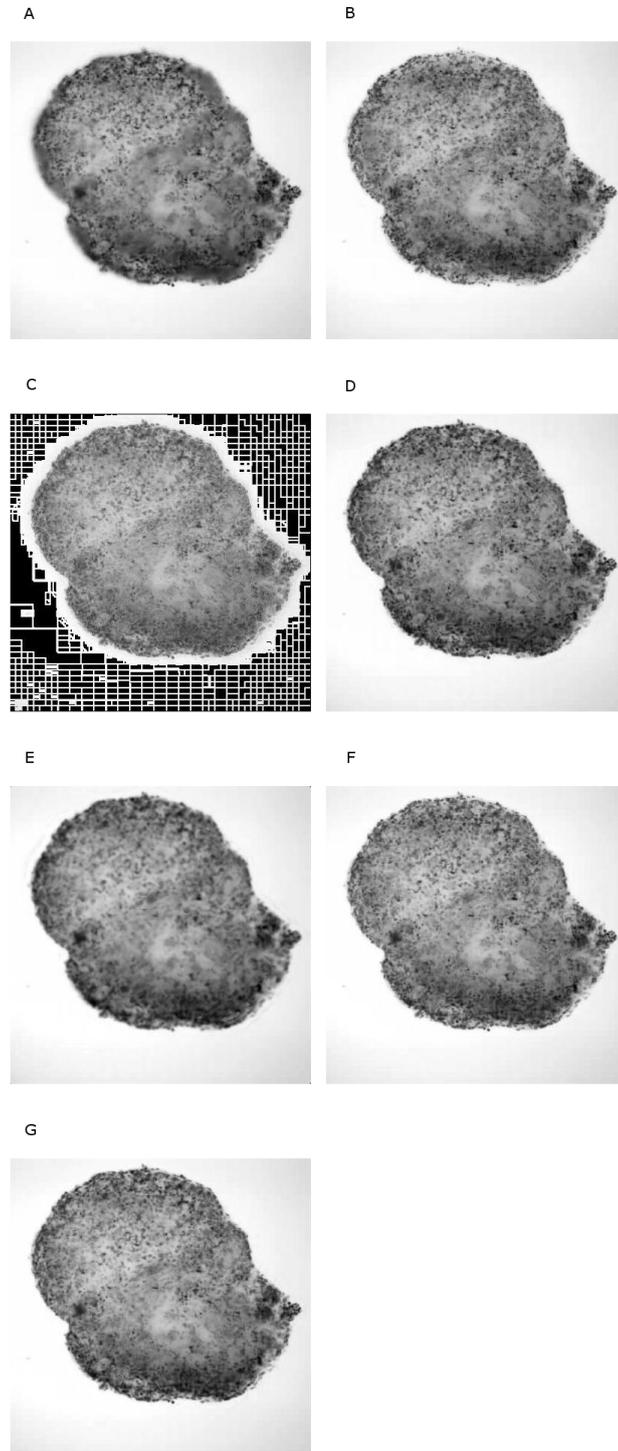


Figura 6.5: Immagini composite finali ottenute usando lo *stack* del processo uncinato del pancreas. Per ottenere le immagini sono stati usati i seguenti algoritmi: (A) MBM, (B) CWT, (C) SI, (D) SF, (E) AnPuLo, (F) DT-CWT, (G) CT.

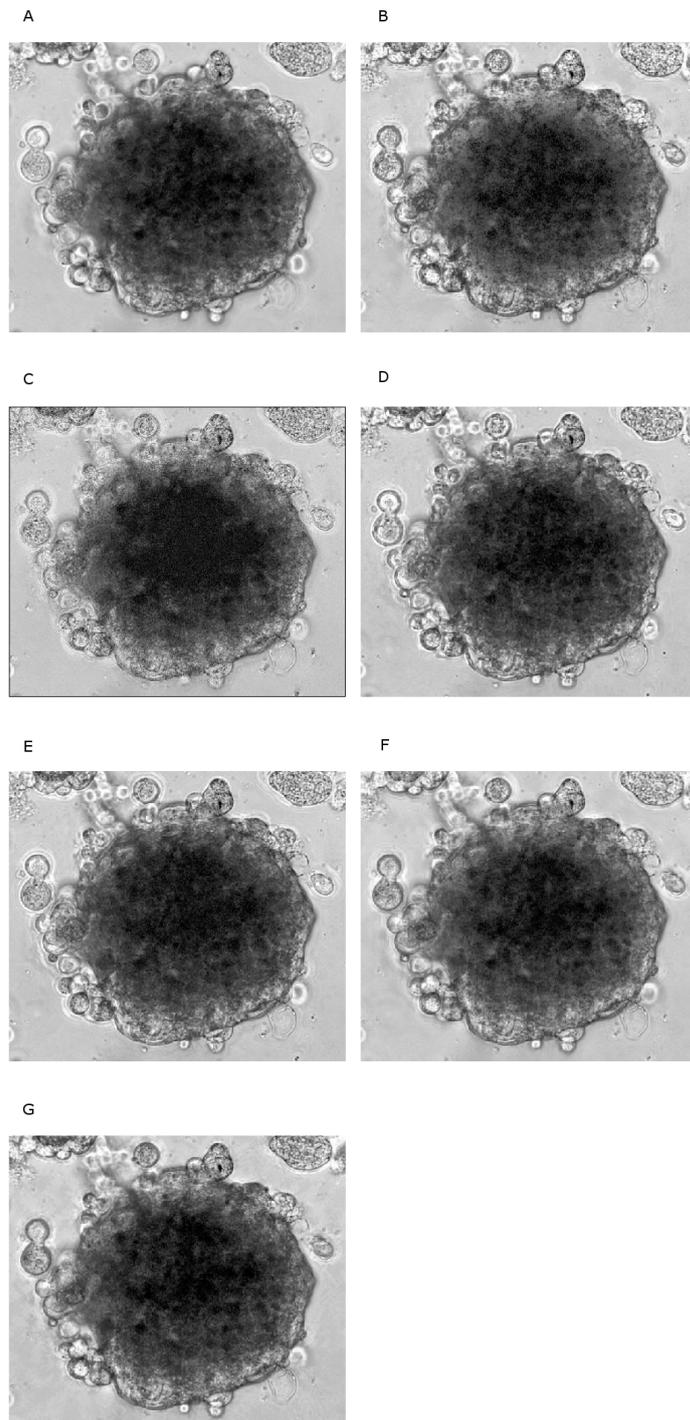


Figura 6.6: Immagini composite finali ottenute usando lo *stack* dello sferoide di cellule polmonari. Per ottenere le immagini sono stati usati i seguenti algoritmi: (A) MBM, (B) CWT, (C) SI, (D) SF, (E) AnPuLo, (F) DT-CWT, (G) CT.

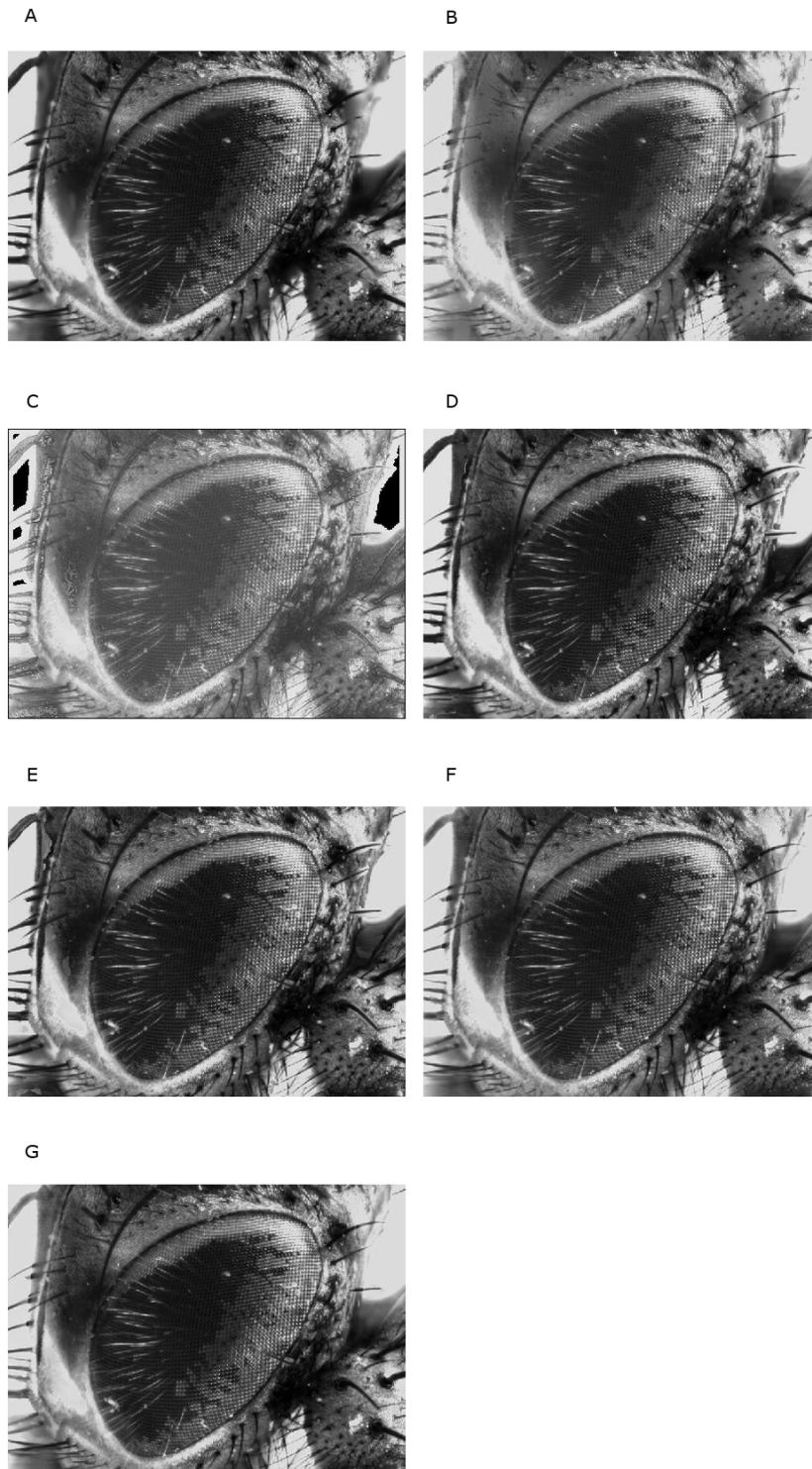


Figura 6.7: Immagini composite finali ottenute usando lo *stack* dell'occhio di mosca.. Per ottenere le immagini sono stati usati i seguenti algoritmi: (A) MBM, (B) CWT, (C) SI, (D) SF, (E) AnPuLo, (F) DT-CWT, (G) CT.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
UQIN Saldatura Laser	0,6576	0,6045	0,4821	0,6014	0,6258	0,6453	0,6468
UQIN _W Saldatura Laser	0,7509	0,7126	0,5879	0,7106	0,7227	0,7428	0,7445
UQIN _E Saldatura Laser	0,5845	0,4547	0,2245	0,5402	0,5570	0,5705	0,5679
UQIN Intestino Topo	0,4656	0,3991	0,3273	0,4045	0,4144	0,4372	0,3902
UQIN _W Intestino Topo	0,6646	0,6119	0,5334	0,6212	0,6233	0,6460	0,6005
UQIN _E Intestino Topo	0,4483	0,3533	0,2468	0,4616	0,4587	0,4465	0,3790
UQIN Pancreas	0,7145	0,6845	0,3482	0,6670	0,6713	0,6930	0,6895
UQIN _W Pancreas	0,7351	0,7059	0,6856	0,698	0,7011	0,7204	0,7246
UQIN _E Pancreas	0,5459	0,5452	0,5391	0,6226	0,6317	0,6311	0,6325
UQIN Sferoide	0,3421	0,3531	0,2738	0,2961	0,2972	0,3833	0,3780
UQIN _W Sferoide	0,4496	0,4642	0,3672	0,4258	0,4096	0,4878	0,5012
UQIN _E Sferoide	0,2509	0,2293	0,1644	0,2634	0,2672	0,273	0,2595
UQIN Occhio Mosca	0,6591	0,5956	0,5746	0,6392	0,6425	0,6384	0,6411
UQIN _W Occhio Mosca	0,7541	0,6980	0,6722	0,7362	0,7327	0,7427	0,7373
UQIN _E Occhio Mosca	0,6995	0,6099	0,6204	0,7161	0,7110	0,7096	0,6974

Tabella 6.11: Valori delle tre versioni dell'UQIN per i data set reali.

	MDB	CWT	SI	SF	AnPuLo	DT-CWT	CT
UQIN Saldatura Laser	1 st	5 th	7 th	6 th	4 th	3 rd	2 nd
UQIN _W Saldatura Laser	1 st	5 th	7 th	6 th	4 th	3 rd	2 nd
UQIN _E Saldatura Laser	1 st	6 th	7 th	5 th	4 th	2 nd	3 rd
UQIN Intestino Topo	1 st	5 th	7 th	4 th	3 rd	2 nd	6 th
UQIN _W Intestino Topo	1 st	5 th	7 th	4 th	3 rd	2 nd	6 th
UQIN _E Intestino Topo	3 rd	6 th	7 th	1 st	2 nd	4 th	5 th
UQIN Pancreas	1 st	4 th	7 th	6 th	5 th	2 nd	3 rd
UQIN _W Pancreas	1 st	4 th	7 th	6 th	5 th	3 rd	2 nd
UQIN _E Pancreas	5 th	6 th	7 th	4 th	2 nd	3 rd	1 st
UQIN Sferoide	4 th	3 rd	7 th	6 th	5 th	1 st	2 nd
UQIN _W Sferoide	4 th	3 rd	7 th	5 th	6 th	2 nd	1 st
UQIN _E Sferoide	5 th	6 th	7 th	3 rd	2 nd	1 st	4 th
UQIN Occhio Mosca	1 st	6 th	7 th	4 th	2 nd	5 th	3 rd
UQIN _W Occhio Mosca	1 st	6 th	7 th	4 th	5 th	2 nd	3 rd
UQIN _E Occhio Mosca	4 th	7 th	6 th	1 st	2 nd	3 rd	5 th

Tabella 6.12: Classifiche delle tre versioni dell'UQIN per i data set reali.

6.6 Metodo di ricostruzione 3-D

Come già detto, l'algoritmo proposto nel Capitolo 3, oltre a generare l'immagine composita finale fornisce in uscita anche la mappa delle profondità. Quest'ultima, oltre ad essere importante per la creazione dell'immagine finale a fuoco, può essere utilizzata anche per ricostruire la superficie 3-D degli oggetti fotografati. In particolare, la mappa della profondità è una matrice

in cui ogni elemento indica il numero della *slice* dello *stack* da cui prelevare il *pixel*, in quella determinata posizione $x-y$, per creare l'immagine finale a fuoco. Si è pensato quindi di usare la *depth map* per ritrovare anche la posizione nello spazio di ogni *pixel* in modo da ottenere una rappresentazione 3-D della superficie dell'oggetto. Siccome in questo caso si va a rappresentare una superficie continua partendo da un'indicazione discreta data da un numero finito di *slice*, si ha l'introduzione di un errore di quantizzazione che è tanto più piccolo quanto più elevato è il numero N di *slice* che compongono lo *stack* (Figura 6.8).

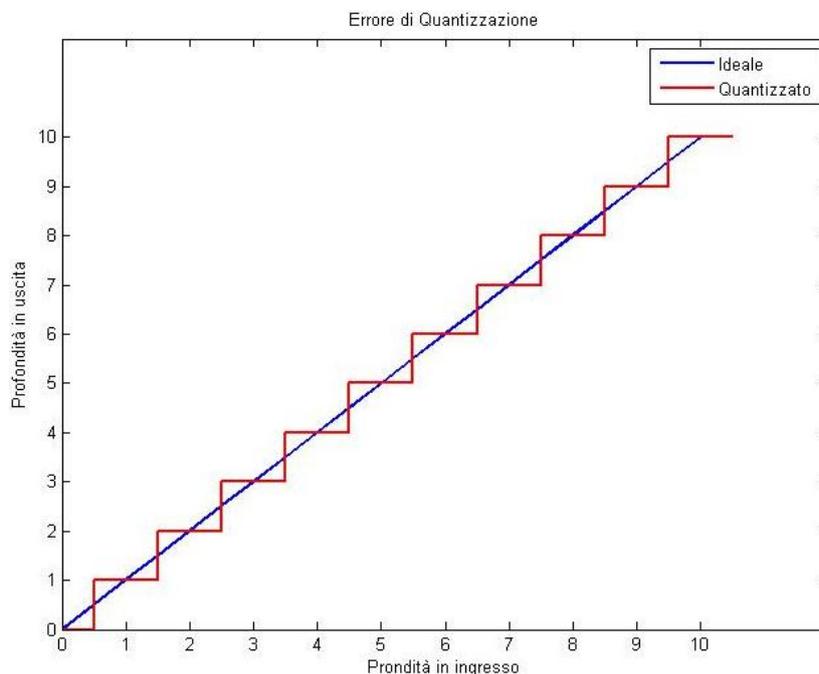


Figura 6.8: Esempio di errore di quantizzazione.

Come effetto di quello appena descritto, si ha che la superficie 3-D ricostruita presenta delle discontinuità dovute al “salto” compiuto nel passare da un valore ad un altro, e questo va ad inficiare il risultato visivo rendendo difficilmente utilizzabile la superficie così ricostruita. Per cercare di risolvere questo problema sono state proposte due distinte strategie: filtraggio a media o *fitting* polinomiale della *depth map*. Con entrambe le tecniche si ottiene visivamente un notevole aumento della qualità della superficie 3-D. Ad occhio, i risultati migliori sono stati ottenuti con il *fitting* polinomiale, ma questa strategia risulta essere maggiormente *object-dependent* rispetto al filtraggio, infatti bisogna ogni volta studiare la superficie reale per capire qual è il grado del polinomio in grado di rappresentare più fedelmente l'andamento dei dati.

Si è scelto quindi di consigliare l'utilizzo del filtraggio a media perché, pur fornendo risultati leggermente inferiori dal punto di vista qualitativo/visivo, risulta essere una soluzione più generale ed indipendente dal contesto.

Capitolo 7

Interfaccia grafica

Come già detto nei capitoli precedenti, il presente lavoro di tesi si inserisce all'interno di un progetto di collaborazione tra l'IRST e il CVG dell'Università di Bologna. In particolare l'algoritmo sviluppato per l'estensione della profondità di campo e la ricostruzione della superficie 3-D dell'oggetto, avrà come utilizzatori finali principalmente dei biologi e dei medici, ai quali interessa disporre di strumenti estremamente *user friendly*. Tenendo conto di questa esigenza, si è pensato di creare un'interfaccia grafica (in inglese *Graphical User Interface*, GUI) che permette di utilizzare l'algoritmo di ricostruzione 3D sviluppando interagendo con oggetti grafici convenzionali (bottoni, caselle di testo e *check-box*). Una applicazione sviluppata su "riga di comando" di una CLI (*command line interface*), in cui l'esecuzione del programma viene guidata da istruzioni impartite tramite la tastiera, in questo caso non risponderebbe alle caratteristiche richieste.

7.1 Strumenti MATLAB per lo sviluppo

L'interfaccia grafica, come gli algoritmi e tutte le funzioni ausiliarie sviluppate per la tesi, è stata creata in ambiente MATLAB © e in particolare si è utilizzato il tool GUIDE (*Graphical User Interface Design Environment*) che fornisce una serie di strumenti per creare interfacce grafiche in modo semplice. GUIDE può essere usato per:

- Progettare un'interfaccia grafica e definirne l'aspetto (*layout*).
- Programmare l'interfaccia grafica, ovvero stabilirne il comportamento in funzione delle azioni dell'utente.

Aperto GUIDE e scegliendo di creare una nuova GUI, il *Layout Editor* crea e visualizza una griglia vuota su cui possono venire allocati i vari componenti (Figura 7.1). A questo punto, si possono definire i componenti che costituiranno la nuova GUI, trascinandoli all'interno della GUI vuota dalla barra dei componenti, dimensionandoli e posizionandoli secondo necessità. Dopo aver definito i componenti della GUI ed averne impostato le proprietà,

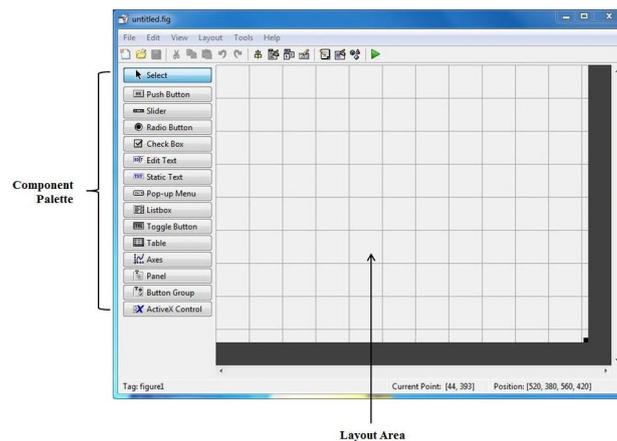


Figura 7.1: Print screen della pagina iniziale del Layout Editor di GUIDE.

il passo successivo è la programmazione della stessa, cioè l'associare funzioni logico/matematiche ai componenti grafici allocati. In parole più semplici, con "programmazione della GUI" si intende la definizione del comportamento della GUI in seguito ad azioni dell'utente sui suoi componenti, e avviene attraverso la scrittura di apposite funzioni MATLAB © associate ai singoli componenti della GUI ed alle possibili azioni dell'utente. Tali funzioni sono in genere chiamate *callback*. GUIDE genera automaticamente un file di testo contenente linee di codice (chiamato M-file), cioè le funzioni *callback* associate alla GUI sviluppata con il *Layout Editor*. Le funzioni generate da GUIDE sono vuote quindi dovranno essere completate dal programmatore con codice adeguato per eseguire le azioni richieste per la specifiche applicazioni.

7.2 Progetto e realizzazione

In Figura 7.2 è mostrata come apparirà all'utente la GUI preparata in questo lavoro di tesi. In particolare, l'interfaccia grafica in questo caso può essere suddivisa in due macro-blocchi principali:

- il **BLOCCO 1**: in cui si posso impostare i parametri e lanciare l'algoritmo AnPuLo per l'estensione della profondità di campo, visualizzando l'immagine completamente a fuoco;
- il **BLOCCO 2** che serve per la visualizzazione della superficie 3-D.

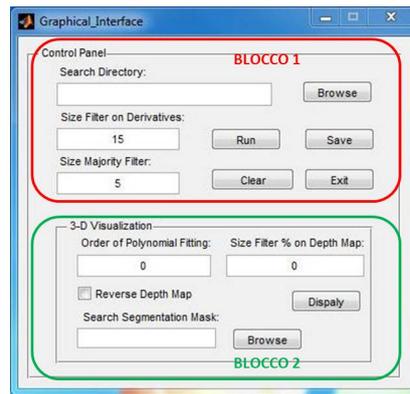


Figura 7.2: Interfaccia grafica.

Analizzando il **BLOCCO 1** della GUI, si può notare che è composto da tre caselle di testo e cinque bottoni. Le due caselle di testo riguardanti i parametri dell'algoritmo, cioè la dimensione del filtro sulle derivate e la dimensione del filtro di maggioranza sono impostati di *default* ai valori ottimali per le applicazioni svolte in questa tesi (rispettivamente 15 e 5, che rappresentano il numero di *pixel* per lato delle finestre quadrate utilizzate nei rispettivi filtraggi). Proprio per il fatto che in altre applicazioni potrebbe essere necessario intervenire su questi parametri, si è lasciata la possibilità di cambiarli andando semplicemente a modificare i valori all'interno delle caselle di testo. Premendo il pulsante "*Browse*" si apre una GUI che permette di scegliere la *directory* dove sono contenute le immagini dello *stack* che si vogliono elaborare (Figura 7.3).

Selezionata la cartella, il percorso (*path*) delle immagini verrà visualizzato nella casella di testo identificata dalla scritta "*Search Directory*" in modo da avere un controllo sulle immagini selezionate prima dell'avvio dell'applicazione. A questo punto attraverso il pulsante "*Run*" è possibile lanciare l'algoritmo AnPuLo e visualizzare l'immagine 2-D completamente a fuoco ottenuta (Figura 7.4).

Le funzioni associate ai tre bottoni del **BLOCCO 1** non ancora descritti sono facilmente intuibile dall'etichette assegnate agli stessi bottoni:

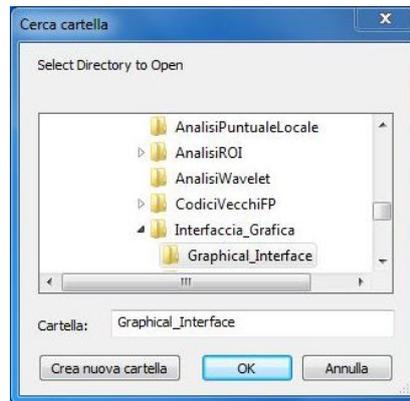
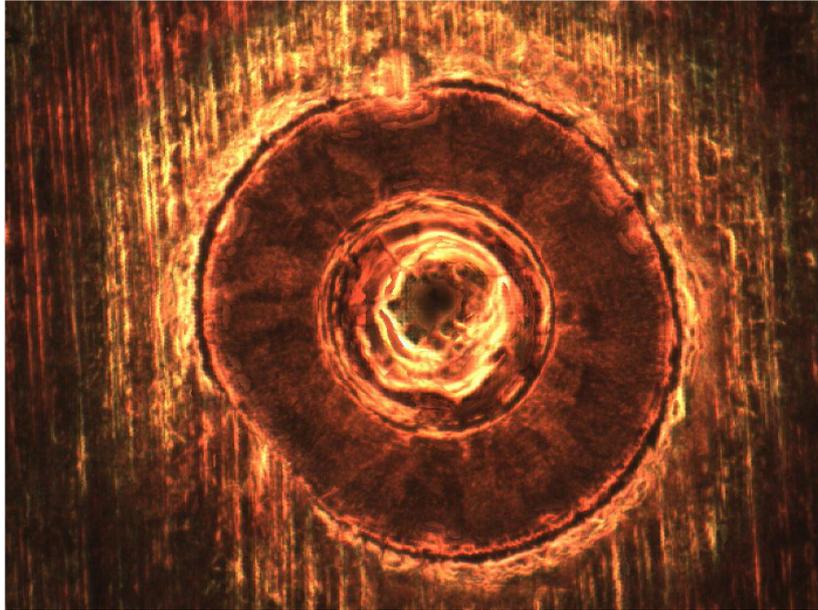


Figura 7.3: Interfaccia grafica che permette di selezionare la cartella in cui sono contenute le immagini.

- *Save*: dopo aver lanciato l'algoritmo, nel *workspace* di MATLAB © sono presenti due variabili che sono rispettivamente l'immagine composita finale a fuoco e la *depth map*. Attraverso il bottone *save* si apre un'interfaccia GUI che permette di salvare in un unico file le due variabili (Figura 7.5).
- *Clear*: elimina tutte le variabili presenti nel *workspace* e chiude tutte le finestre aperte in MATLAB © riportando tutti i parametri dell'interfaccia ai valori di *default*.
- *Exit*: permette di uscire dalla GUI.

Passiamo quindi a descrivere il **BLOCCO 2**. In particolare, quest'ultimo può essere usato solo se è già stato lanciato l'algoritmo, altrimenti viene visualizzato un messaggio di errore perché, come già detto, questo blocco serve solo per visualizzare la superficie 3-D e necessita che siano già state create l'immagine composita finale e la *depth map*. Il fulcro di questo blocco sono le due caselle di testo etichettate come "*Order of Polynimial Fitting*" e "*Size Filter % on Depth Map*", e il bottone "*Display*". In particolare, le due caselle di testo servono per impostare i parametri dei due metodi per la ricostruzione della superficie 3-D descritti nel Capitolo 6, e di *default* sono entrambi settati a zero. Il valore all'interno della casella definita "*Order of Polynimial Fitting*" rappresenta il grado del polinomio con il quale si "fitta" la superficie reale dell'oggetto; mentre il valore della casella "*Size Filter % on Depth Map*" indica il lato della finestra quadrata usata per il filtraggio a media, ed è definito sotto forma di percentuale della dimensione maggiore dell'immagine (es. $Dim.Immagine = [1024, 768]$, $Size Filter \% on Depth Map =$

A



B



Figura 7.4: Esempi di immagini composite finali ottenute lanciando l'algoritmo attraverso l'interfaccia. (A) Placche di Peyer dell'intestino di topo; (B) Saldatura Laser.

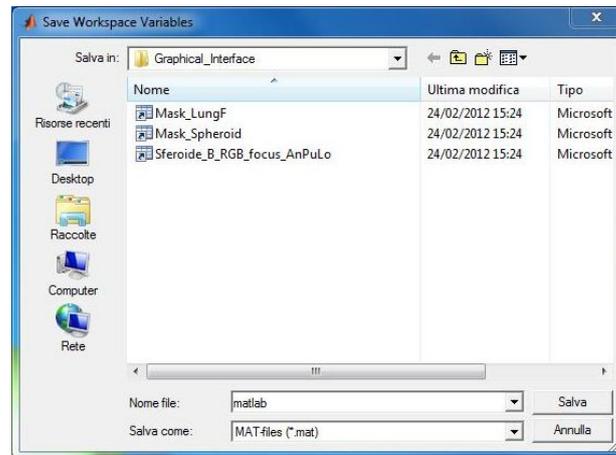


Figura 7.5: Interfaccia grafica per il salvataggio delle variabili presenti nel *workspace*.

10, lato finestra quadrata = $(\max(1024, 768) * 10) / 100 = 102,4 \approx 103\text{pixel}$, il valore è approssimato all'intero dispari maggiore più vicino). In Figura 7.6 e in Figura 7.7 sono mostrati alcuni esempi di ricostruzione della superficie 3-D ottenuti sia con il metodo del *fitting* polinomiale sia con il metodo del filtraggio.

Infine, descriviamo gli ultimi componenti aggiuntivi allocati sull'interfaccia grafica. La *check-box* se attivata, serve per invertire la mappa delle profondità. Può essere utile per visualizzare al meglio la superficie in quanto se le immagine dello *stack* sono state acquisite dall'alto verso il basso, alle immagini più profonde viene assegnato un valore maggiore e quindi la superficie così ricostruita sarebbe capovolta. La casella di testo etichettata come "*Search Segmentation Mask*" e il suo relativo bottone "*Browse*" servono per caricare una maschera per la segmentazione della *depth map* (Figura 7.9, Figura 7.10). È stata introdotta questa possibilità perché cercando di costruire il grafico 3-D della superficie dello sferoide di cellule polmonari, si è visto che essendo la *depth map* molto rumorosa ai bordi, non segmentando il *foreground* (lo sferoide principale) dal *background* si otteneva una superficie difficilmente decifrabile e utilizzabile.

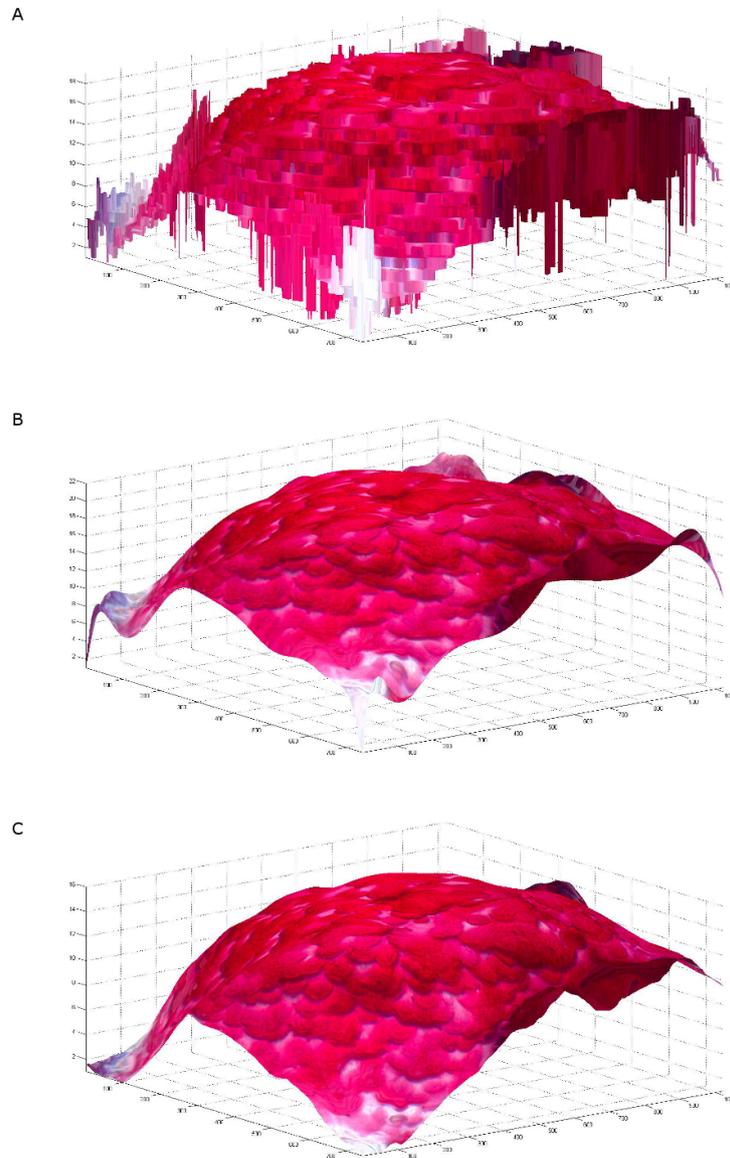


Figura 7.6: (A) Ricostruzione della superficie 3-D dello *stack* delle placche Peyer dell'intestino di topo utilizzando la *depth map* senza nessun *post-processing*. (B) *Fitting* con un polinomio di 15° grado della topografia dell'immagine A. (C) Filtraggio a media della superficie in A con *kernel* di dimensioni pari al 10% della dimensione maggiore dell'immagine.

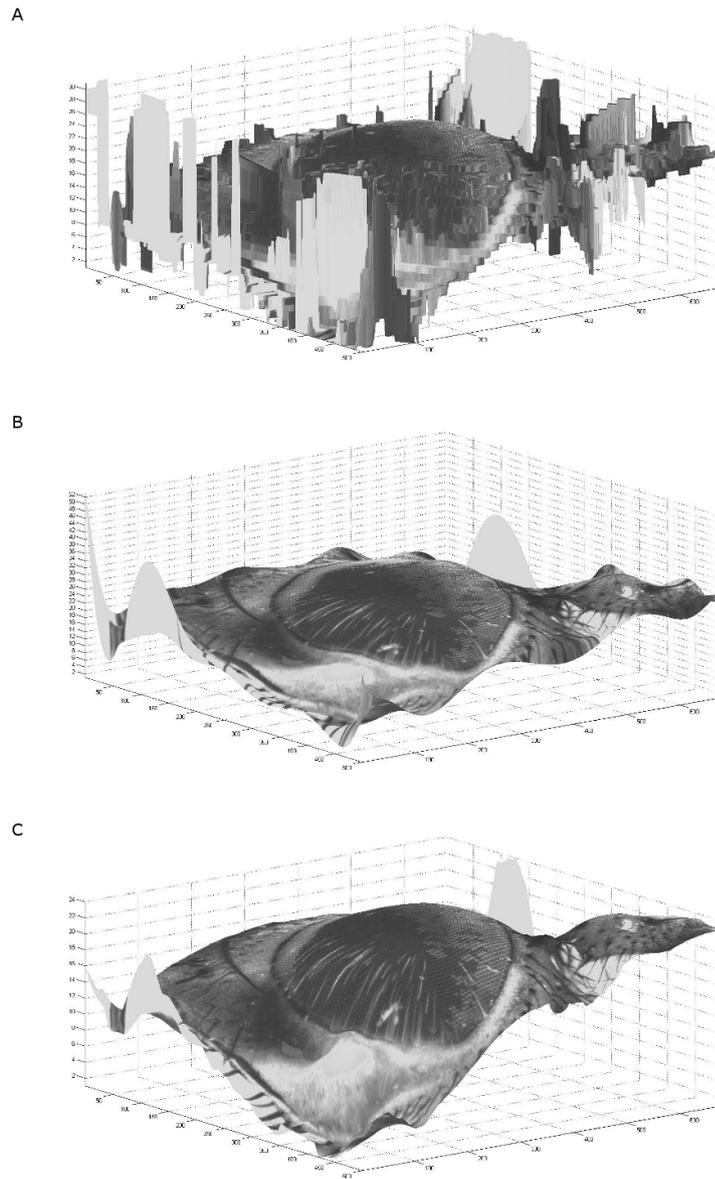


Figura 7.7: (A) Ricostruzione della superficie 3-D dello *stack* dell'occhio di mosca utilizzando la *depth map* senza nessun *post-processing*. (B) *Fitting* con un polinomio di 15° grado della topografia dell'immagine A. (C) Filtraggio a media della superficie in A con un *kernel* di dimensioni pari al 10% della dimensione maggiore dell'immagine.



Figura 7.8: Immagine finale composita a fuoco dello sferoide di cellule polmonari.

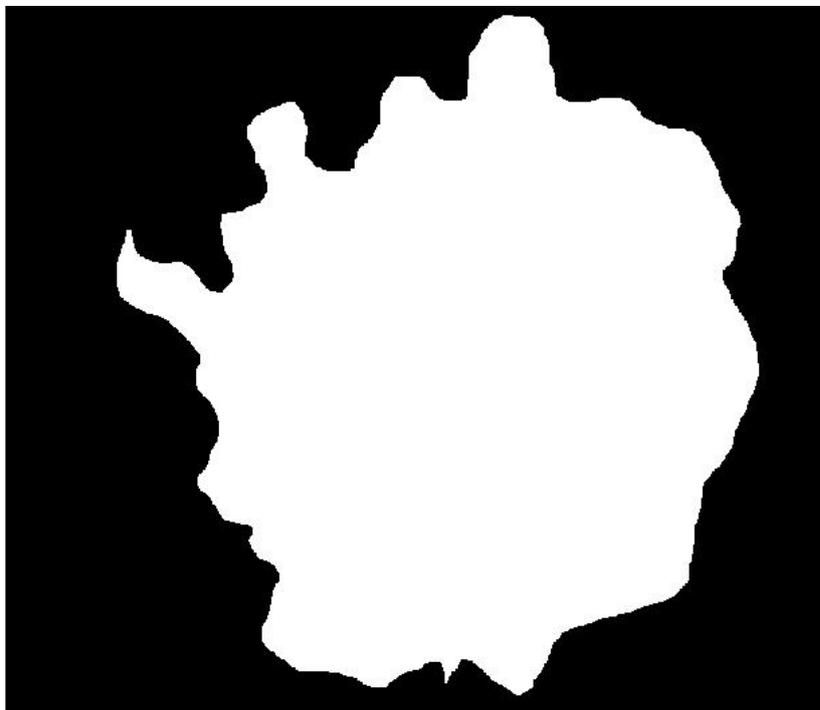


Figura 7.9: Maschera di segmentazione per la *depth map* dello sferoide.

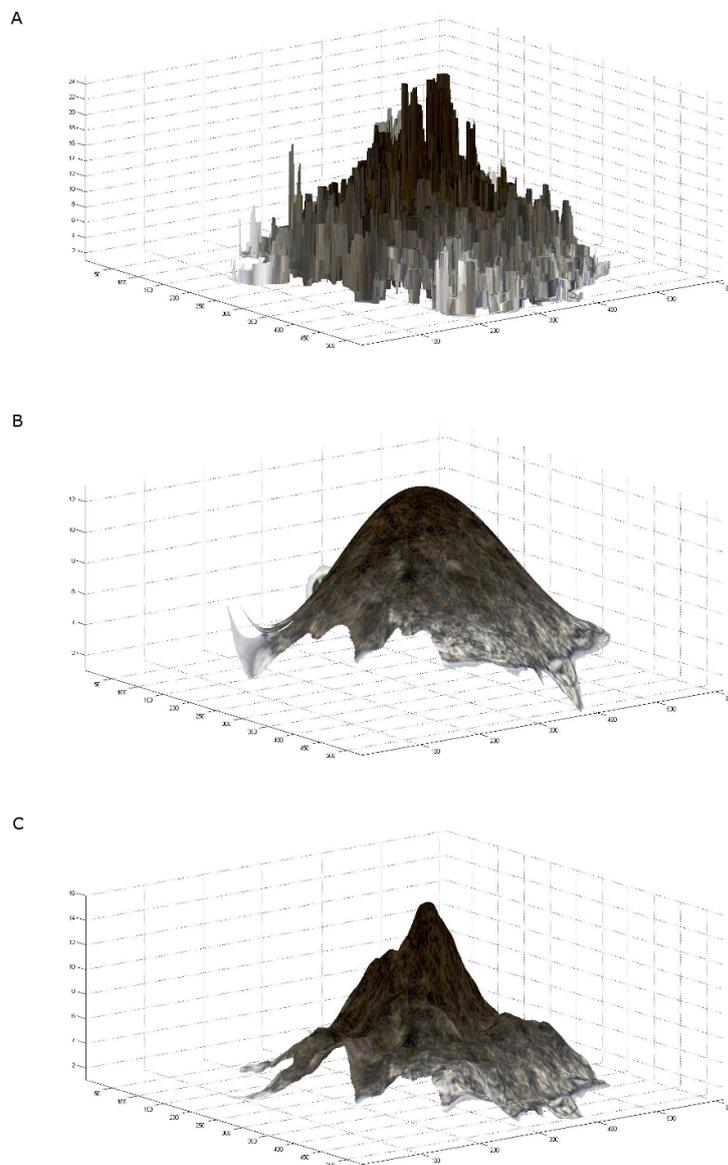


Figura 7.10: **(A)** Ricostruzione della superficie 3-D dello *stack* dello sferoide utilizzando la *depth map* senza nessun *post-processing*. **(B)** *Fitting* con un polinomio di 15° grado della topografia dell'immagine A. **(C)** Filtraggio a media della superficie in A con un *kernel* di dimensioni pari al 10% della dimensione maggiore dell'immagine.

Capitolo 8

Conclusioni e sviluppi futuri

Il lavoro svolto in questa tesi si inserisce in una consolidata collaborazione tra il Computer Vision Group (CVG), coordinato dal Prof. Bevilacqua ed il Laboratorio di Bioscienze dell'Istituto scientifico Romagnolo per lo Studio e la cura dei Tumori (IRST), coordinato dal Dott. Wainer Zoli. L'estensione della profondità di fuoco è un problema ben noto nella microscopia ottica convenzionale. Tipicamente è impossibile acquisire una singola immagine completamente a fuoco di oggetti caratterizzati da dimensioni molto elevate, maggiori della profondità di fuoco del sistema. In particolare, tra questi oggetti sono incluse le broncosfere studiate e prodotte ad IRST, essendo sferoidi multicellulari che possono raggiungere dimensioni molto grandi (1 mm di diametro e anche oltre). Molti metodi sono stati proposti in letteratura per risolvere questo problema, ma a causa della mancanza della *ground truth* e di un approccio validato per misurare le performance ottenute risulta difficile decidere quale metodo potrebbe portare i risultati migliori in una specifica applicazione. Tuttavia i metodi operanti nel dominio delle frequenze, basati sull'utilizzo delle trasformate wavelet e curvelet, sono largamente utilizzati e spesso citati in quanto migliori rispetto agli approcci che operano nel dominio spaziale. Nonostante ciò, la semplicità e la gestibilità degli approcci nel dominio spaziale rimangono molto attraenti. Partendo da ciò, in questo lavoro di tesi è stato proposto un nuovo metodo di estensione della profondità di fuoco in grado di ricostruire l'immagine finale completamente a fuoco partendo da una serie di immagini acquisite a diverse lunghezze focali lavorando nel dominio spaziale. In particolare, le regioni a fuoco nelle singole immagini originali sono identificate attraverso una semplice derivata approssimata con le differenze finite centrali e l'immagine a fuoco è creata utilizzando una regola di selezione dei massimi della derivata e alcuni stadi

di filtraggio. Alcuni stadi di filtraggio aggiuntivi si sono poi dimostrati determinanti per ottenere un'elevata qualità dei risultati prodotti.

Per valutare l'efficacia delle prestazioni ottenute con il metodo proposto, queste sono state confrontate con quelle ottenute da sei altri differenti algoritmi, ampiamente usati e citati in letteratura. Gli esperimenti eseguiti usando *data set* sintetici, corredati di *ground truth*, hanno provato che le prestazioni ottenute dall'algoritmo proposto in questo lavoro di tesi, risultano migliori di quelle dei metodi allo stato dell'arte. Per poter verificare come l'algoritmo si comporta nei casi reali, come prima cosa è stato dimostrato sfruttando le immagini sintetiche correlate da *ground truth* che alcune metriche proposte in letteratura come estensioni dell'UQI *standard* non sono in grado in realtà di classificare i metodi testati come farebbe l'UQI. Quindi, è stato proposto e validato un nuovo approccio di classificazione basato sulla stima della media dei valori dell'UQI, calcolati tra l'immagine composita finale ottenuta dall'algoritmo e ogni singola *slice* dello *stack* di ingresso. È stato dimostrato che questo approccio è capace di classificare gli algoritmi per l'estensione della profondità di campo allo stesso modo dell'UQI *standard*. È stato allora deciso di utilizzare questa metrica per studiare come il metodo proposto in questa tesi si comporta con *data set* di immagini reali, che ovviamente non sono provviste di *ground truth*. I risultati ottenuti hanno dimostrato che le prestazioni dell'algoritmo proposto, nonostante la semplicità, sono superiori o quanto meno paragonabili a quelle ottenute con gli algoritmi allo stato dell'arte. Inoltre, è stato dimostrato che, al contrario di quanto previsto in molti lavori, non è vero che gli approcci nel dominio spaziale sono comunque peggiori di quelli basati sulle wavelet o sulle curvelet.

Molte questioni restano tuttora aperte. In particolare:

- non esiste una metodologia valida adatta per decretare quale algoritmo sia il migliore confrontando immagini a colori ottenute dai diversi metodi per l'estensione della profondità di campo. Un approccio potrebbe essere quello di trasformare le immagini in scala di grigio solo per la valutazione della qualità, ma in questo modo problemi quali la presenza di falsi colori o cambiamenti di tonalità non vengono tenuti correttamente in considerazione
- l'algoritmo proposto potrebbe essere sicuramente migliorato dal punto di vista della efficienza computazionale e dell'utilizzo della memoria per lo *storage* dei dati utilizzati in fase di valutazione della qualità

- la ricostruzione della superficie 3-D può essere ampiamente migliorata. In particolare, si potrebbe pensare di ricostruire in 3-D tutto l'oggetto affiancando alle informazioni date dalla mappa delle profondità, quelle inerenti alla densità derivabili da un legame tra tono di grigio e densità.

Comunque, con mia grande soddisfazione personale, questo lavoro di Tesi ha originato un primo lavoro già sottomesso in sede internazionale:

Filippo Piccinini, Anna Tesei, Wainer Zoli, Ludovico Carozza, Davide Pollini, Alessandro Bevilacqua, "*Extending the Universal Quality Index to assess N-image fusion in optical microscopy*", submitted to: 7th International Workshop on Biosignal Interpretation (BSI2012), July 2-4 2012, Como (Italy).

Appendice A

Trasformata Wavelet e Trasformata Curvelet

Per poter comprendere i dettagli dei metodi per ricostruzione basati sulla decomposizione multiscala, è necessario ripassare alcuni concetti fondamentali relativi alle wavelet. Sin dalla loro definizione, circa 30 anni fa, le trasformate wavelet sono state utilizzate in vari campi applicativi riguardanti l'elaborazione di segnali e hanno ottenuto un gran successo spesso contribuendo a progressi dello stato dell'arte. La trasformata wavelet discreta (in inglese *discrete wavelet transform*, DWT) è definita partendo dalla trasformata di Fourier, usando un set di funzioni base oscillanti a lunghezza finita chiamate wavelet per sostituire le funzioni base sinusoidali. Nella classica rappresentazione, le differenti wavelet utilizzate non sono altro che una versione scalata e traslata di una wavelet madre $\psi(t)$. Accuratamente combinate con funzioni di scala passa-basso $\phi(t)$, formano una base ortonormale per un segnale monodimensionale tempo continuo. Quindi per definizione di base ortonormale, un segnale $x(t)$ ad energia finita può essere così decomposto in wavelet e funzioni di scala:

$$x(t) = \sum_{n=-\infty}^{\infty} c(n)\phi(t-n) + \sum_{j=0}^{\infty} \sum_{n=-\infty}^{\infty} d(j,n)2^{j/2}\psi(2^j t - n) \quad (\text{A.1})$$

I coefficienti di scala $c(n)$ e i coefficienti wavelet $d(j,n)$ sono calcolati attraverso il prodotto scalare:

$$c(n) = \int_{-\infty}^{\infty} x(t)\phi(t-n)dt \quad (\text{A.2})$$

$$d(j, n) = 2^{j/2} \int_{-\infty}^{\infty} x(t)\psi(2^j t - n)dt \quad (\text{A.3})$$

Questi coefficienti forniscono un'analisi tempo-frequenza del segnale misurando il suo contenuto frequenziale, tramite il fattore di scala j , a diversi istanti di tempo, controllati dalla traslazione temporale n . Esiste un algoritmo lineare, molto efficiente, usato per calcolare i coefficienti $c(n)$ e $d(j, n)$ da una rappresentazione ad alta frequenza del segnale e basato su dei banchi di filtri a tempo discreto (in inglese *discrete-time filter banks*) che ricorsivamente applicano un filtraggio passa-basso $h_0(n)$, un filtraggio passa-alto $h_1(n)$, e operazioni di *downsampling* e *upsampling* (Figura A.1).

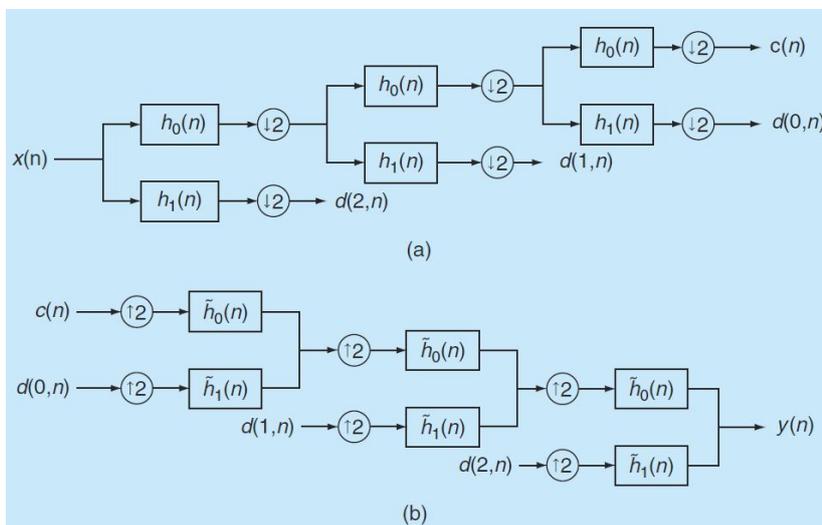


Figura A.1: (a)Banchi di filtri per la DWT e (b) per la DWT inversa . Cortesemente da [42].

Questi filtri forniscono una utile parametrizzazione per costruire le funzioni wavelet e le funzioni di scala con proprietà desiderate, come durata finita e veloce frequenza di decadimento, necessarie affinché l'analisi risulti il più locale possibile nel dominio tempo-frequenza. La trasformata wavelet presenta purtroppo quattro difetti altamente correlati tra loro:

1. Oscillazioni: siccome le wavelet sono funzioni passa-banda, i loro coefficienti tendono ad oscillare positivamente e negativamente intorno a delle singolarità.
2. Varianza alla traslazione: un piccolo spostamento del segnale perturba in maniera molto accentuata l'oscillazione del coefficiente intorno alle singolarità e ciò complica l'elaborazione nel dominio delle wavelet:

gli algoritmi devo essere in grado di far fronte alla vasta gamma di possibili modelli dei coefficienti wavelet dovuti alla traslazione delle singolarità.

3. *Aliasing*: dato che i coefficienti wavelet sono calcolati attraverso processi iterativi a tempo discreto di operazioni di *downsampling* intervallate da filtri passa-basso e passa-alto non ideali, si ha come risultato un sostanziale effetto di *aliasing*. La DWT inversa cancella l'*aliasing* ma solamente se i coefficienti wavelet e di scala non sono stati modificati. Qualsiasi modifica dei coefficienti perturba il delicato equilibrio tra la trasformata e l'antitrasformata portando alla formazioni di artefatti nel segnale ricostruito.
4. Perdita di direzionalità: il prodotto tensoriale *standard* di wavelet multidimensionali produce uno schema a scacchiera che è allo stesso tempo orientato in diverse direzioni. Questa mancanza selettiva di direzionalità complica l'elaborazione di alcune caratteristiche di immagini geometriche come i picchi e i contorni.

Fortunatamente esiste una semplice soluzione a questi quattro difetti. La chiave sta nel notare che la trasformata di Fourier non è soggetta a questi problemi; infatti a differenza della DWT, che utilizza delle wavelet a valori reali, la trasformata di Fourier è basata su sinusoidi a valori complessi:

$$e^{j\Omega t} = \cos(\Omega t) + j \sin(\Omega t) \quad (\text{A.4})$$

Inspirandosi alla rappresentazione di Fourier, si può immaginare una trasformata wavelet complessa come nelle Equazioni (A.1),(A.2),(A.3) ma con funzioni di scala a valori complessi e con wavelet a valori complessi:

$$\psi_c(t) = \psi_r(t) + j\psi_i(t) \quad (\text{A.5})$$

Inoltre, se $\psi_r(t)$ e $\psi_i(t)$ formano una coppia di funzioni che soddisfano la trasformata di Hilbert, allora $\psi_c(t)$ risulta essere un segnale analitico ed è possibile rappresentarlo sfruttando il solo asse positivo delle frequenze grazie alla simmetria hermitiana. Proiettando il segnale su $2^{j/2}\psi_c(2^j t - n)$ come nella A.3 si ottiene il coefficiente wavelet complesso:

$$d_c(j, n) = d_r(j, n) + jd_i(j, n) \quad (\text{A.6})$$

Come per la trasformata di Fourier, la trasformata wavelet complessa può essere utilizzata per analizzare e rappresentare sia segnali reali, sia segnali a valori complessi [42].

In realtà qualsiasi CWT basata su wavelet di durata finita non soddisfa appieno le proprietà della trasformata di Hilbert affinché un segnale sia analitico, questo significa che i quattro problemi presenti nella DWT non sono completamente risolti. Inoltre si è visto che risulta essere molto difficile costruire dei filtri che permettano una perfetta ricostruzione del segnale e allo stesso tempo producano delle wavelet analitiche.

Per risolvere questi problemi un nuovo approccio è stato introdotto da Kingsbury nel 1998 e è stato chiamato trasformata wavelet complessa a doppio albero (in inglese *Dual-Tree Complex Wavelet Transform*, DT-CWT). La DT-CWT implementa due DWT reali; la prima DWT genera la parte reale della trasformata, mentre la seconda fornisce la parte immaginaria (Figura A.2).

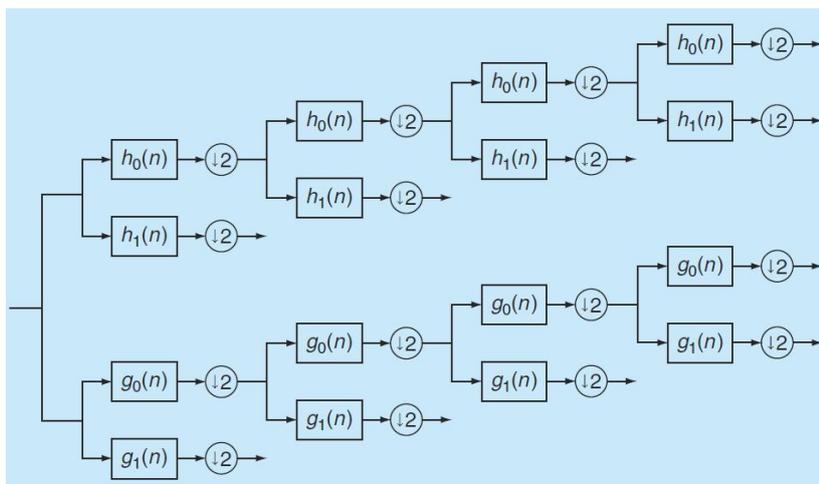


Figura A.2: Banco di filtri per l'analisi per la DT-CWT. Cortesemente da [42].

Le due trasformate usano due set di filtri differenti, e ognuno soddisfa le condizioni per la perfetta ricostruzione. I due set di filtri sono progettati in maniera congiunta in modo che la trasformazione complessiva risulti essere approssimativamente analitica. Indicando le due wavelet reali associate alle trasformazioni con $\psi_g(t)$ e $\psi_h(t)$, allora per soddisfare la condizione di perfetta ricostruzione del segnale, i filtri sono costruiti in modo che la funzione wavelet complessa $\psi(t) = \psi_h(t) + j\psi_g(t)$ sia approssimativamente analitica che equivale a dire che $\psi_g(t)$ sia la trasformata di Hilbert della funzione $\psi_h(t)$. Inoltre i filtri sono reali, nessun calcolo complesso è richiesto

per l'implementazione della DT-CWT. È emerso da studi che per far sì che l'implementazione della DT-CWT sia approssimativamente analitica in tutti gli stadi è necessario che il primo stadio di filtri sia diverso dai successivi [42]. Estendendo al trattamento alle due dimensioni (2-D), si può subito capire perché la 2-D DT-CWT risulti essere molto più efficiente nella rappresentazione dei contorni rispetto alla 2-D DWT.

La 2-D DWT è caratterizzata da tre wavelet:

$$\psi_1(x, y) = \phi(x)\psi(y) \quad (\text{Low} - \text{High}(LH) \text{ wavelet}) \quad (\text{A.7})$$

$$\psi_2(x, y) = \psi(x)\phi(y) \quad (\text{High} - \text{Low}(HL) \text{ wavelet}) \quad (\text{A.8})$$

$$\psi_3(x, y) = \psi(x)\psi(y) \quad (\text{High} - \text{High}(HH) \text{ wavelet}) \quad (\text{A.9})$$

dove ϕ e ψ sono rispettivamente le funzioni passa-basso e passa-alto. Mentre le wavelet LH e HL sono orientate verticalmente e orizzontalmente, la wavelet HH ha un aspetto a scacchiera in quanto mischia le orientazioni a $\pm 45^\circ$. Questo è dovuto al fatto che sia $\psi(x)$, sia $\psi(y)$ sono funzioni reali e quindi il loro spettro è bilatero perciò questa wavelet non è in grado di distinguere fra caratteristiche nel dominio spettrale orientate a $+45^\circ$ e -45° (Figura A.3) e, questo porta alla stessa ambiguità anche nel dominio spaziale.

Le wavelet generate dalla DT-CWT sono orientate, questo perché sia ϕ , sia ψ sono funzioni complesse approssimativamente analitiche e ciò comporta che lo spettro di una wavelet complessa in due dimensioni è presente solo in un quadrante del piano delle frequenze. Prendendo solo la parte reale della wavelet complessa, essendo una funzione reale, il suo spettro risulta essere simmetrico rispetto all'origine del piano e quindi risulta essere presente in due quadranti. Perciò con l'utilizzo della DT-CWT si riescono a distinguere sei orientazioni nel dominio spettrale e quindi sei orientazioni nel dominio spaziale, $\pm 15^\circ, \pm 45^\circ$ e $\pm 75^\circ$ (Figura A.4).

Con l'introduzione delle wavelet complesse si è ottenuto una selettività direzionale sicuramente migliore rispetto alle wavelet reali ma ancora limitata.

Nel 1999, venne proposta da Candès e Donoho una trasformata wavelet geometrica anisotropica, chiamata trasformata ridgelet. La trasformata ridgelet è ottima per la rappresentazione delle singolarità lineari, ma sfortunatamente questo tipo di singolarità è poco presente in applicazioni reali.

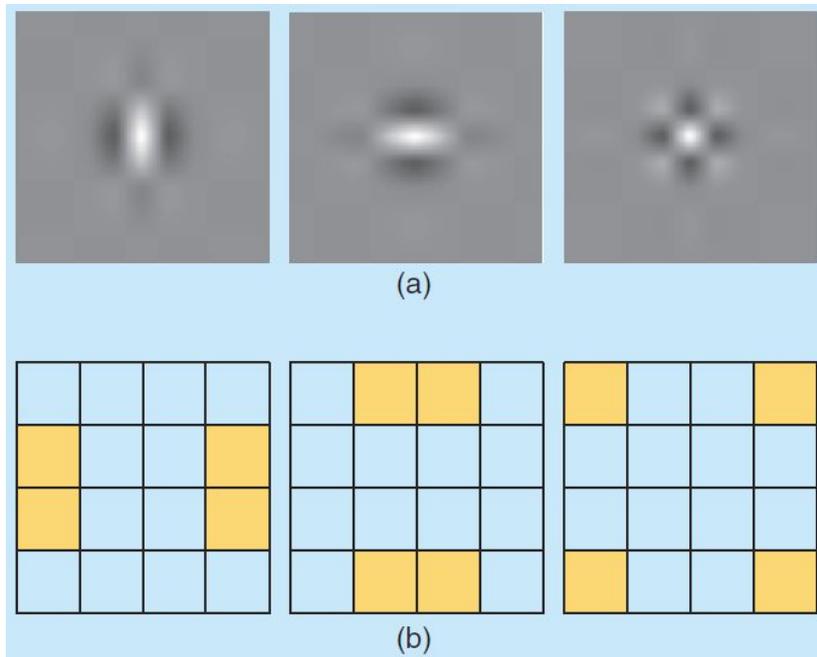


Figura A.3: (a) Wavelet nel dominio spaziale (LH, HL, HH); (b) spettro di Fourier ideale per ciascuna wavelet. Cortesemente da [42].

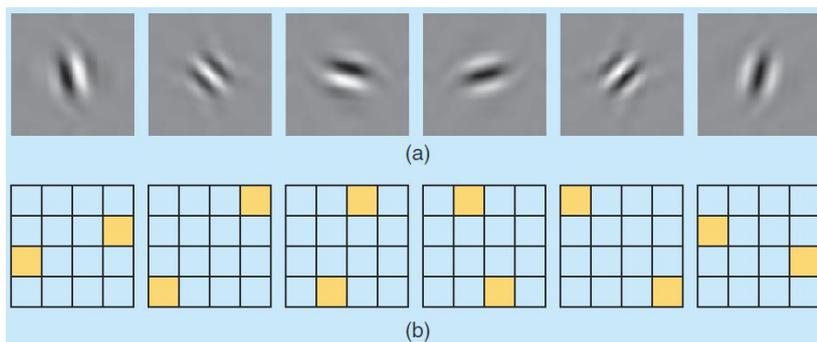


Figura A.4: (a) Rappresentazione spaziale delle sei orientazioni ottenute tramite la DT-CWT; (b) spettro di Fourier ideale delle wavelet ($-75^\circ, -45^\circ, -15^\circ, +15^\circ, +45^\circ, +75^\circ$). Cortesemente da [42].

Per analizzare singolarità locali e curve, un'idea sempre proposta da Cendès e Donoho, fu di considerare una partizione dell'immagine e poi applicare la trasformata ridgelet per ottenere sotto-immagini. Questo nuovo approccio venne chiamato trasformata curvelet (CT). Tuttavia, questa prima generazione di CT non fu molto usata a causa della natura geometrica poco chiara delle funzioni ridgelet. Successivamente fu proposta, sempre dagli stessi autori, una seconda generazione di CT, basata su una tecnica di partizione dello spazio delle frequenze.

La CT discreta risulta essere molto efficiente nella rappresentazione dei contorni ma presenta due principali inconvenienti:

1. le curvelet non sono ottimali per la rappresentazione sparsa di caratteristiche curve per spazi vettoriali delle singolarità di ordine superiore al secondo,
2. la CT discreta è altamente ridondante [32].

Per costruire le curvelet si devono utilizzare le coordinate polari (r, θ) per il dominio delle frequenze e considerare due finestre $W(r)$ e $V(t)$ definite rispettivamente la “finestra radiale” e la “finestra angolare”. Per ogni $j \geq j_0$ si può introdurre una finestra di frequenza U_j definita nel dominio di Fourier come:

$$U_j = 2^{-3j/4} W(2^{-j}r) V\left(\frac{2^{\lfloor j/2 \rfloor} \theta}{2\pi}\right) \quad (\text{A.10})$$

dove $\lfloor j/2 \rfloor$ è la norma (parte intera) di $j/2$. Perciò, la regione in cui è contenuta la funzione U_j è un “cuneo” polare (“*polar wedge*”) definito dalle finestre W e V di larghezza dipendente dalla scala in ogni direzione. Per ottenere curvelet a valori reali bisogna rendere simmetrica la l'Equazione (A.10), cioè, $U_j(r, \theta) + U_j(r, \theta + \pi)$. Si definisce la forma d'onda ψ_j attraverso la sua trasformata di Fourier $F\{\psi_j\} = U_j$. Si può pensare a ψ_j come una curvelet “madre” nel senso che tutte le curvelet alla scala 2^{-j} sono ottenute attraverso una traslazione e una rotazione di ψ_j (Figura A.5). Introducendo:

- la sequenza equispaziata di angoli di rotazione $\theta_l = 2\pi \cdot 2^{-\lfloor j/2 \rfloor} \cdot l$ con $l = 0, 1, \dots$ tale che $0 \leq \theta_l < 2\pi$
- e la sequenza dei parametri di traslazione $k = (k_1, k_2) \in Z^2$.

Si definiscono curvelet alla scala 2^{-j} , orientazione θ_l e posizione $x_k^{(j,l)} = R_{\theta_l}^{-1}(k_1 \cdot 2^{-j}, k_2 \cdot 2^{-j/2})$ le seguenti funzioni:

$$\psi_{j,l,k}(x) = \psi_j(R_{\theta_l}(x - x_k^{j,l})) \quad (\text{A.11})$$

Dove R_θ è la rotazione di θ radianti e R_θ^{-1} è la sua inversa (anche la sua trasposta),

$$R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}, R_\theta^{-1} = R_\theta^T = R_{-\theta} \quad (\text{A.12})$$

I coefficienti curvelet sono definiti come il prodotto scalare tra una funzione f e una curvelet $\psi_{j,l,k}$:

$$c_{j,l,k} = \langle f, \psi_{j,l,k} \rangle = \int_{\mathbb{R}^2} f(x) \overline{\psi_{j,l,k}(x)} dx \quad (\text{A.13})$$

Si deve tenere conto anche degli elementi della scala più grossolana; per fare questo si introduce una finestra passa-basso W_0 e per $k_1, k_2 \in \mathbf{Z}$, si definiscono curvelet della scala grossolana come:

$$\psi_{j_0,k} = \psi_0(x - 2^{j_0}k), F\psi_{j_0}(\omega) = 2^{-j_0}W_0(2^{-j_0}|\omega|) \quad (\text{A.14})$$

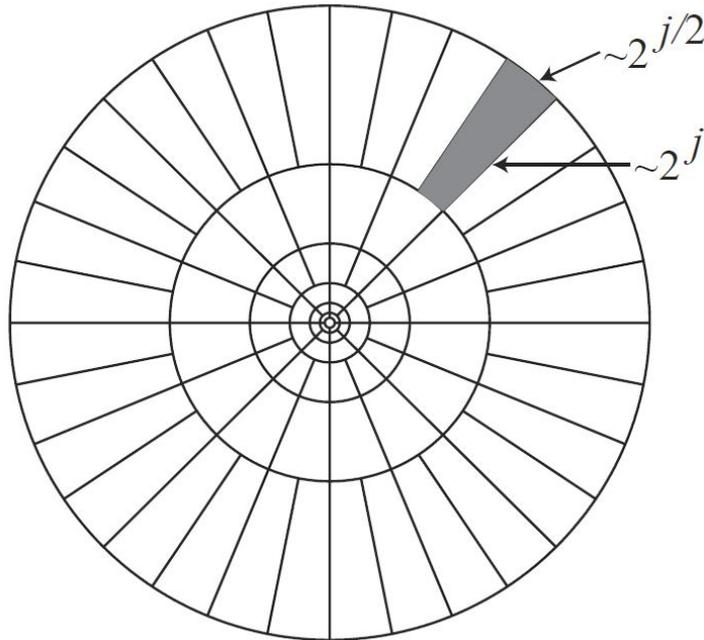


Figura A.5: Rappresenta la suddivisione introdotta nello spazio delle frequenze dalle curvelet. L'area colorata rappresenta un generico cuneo (*wedge*). Cortesemente da [10].

Ovviamente è possibile scrivere una formula analoga all'Equazione (A.13)

per poter calcolare i coefficienti curvelet anche nel caso di segnali digitali:

$$c_{j,l,k}^D = \sum_{0 \leq t_1, t_2 < n} f[t_1, t_2] \overline{\psi_{j,l,k}^D[t_1, t_2]} \quad (\text{A.15})$$

Ogni $\psi_{j,l,k}^D$ è una funzione curvelet digitale.

Nella definizione tempo-continua dell'Equazione (A.10), la finestra U_j estrae le frequenze vicine alla corona diadica $\{2^j \leq r \leq 2^{j+1}\}$, e vicine all'angolo $\{-\pi \cdot 2^{-j/2} \leq \theta \leq \pi \cdot 2^{-j/2}\}$. Corone circolari e rotazioni non sono molto adatte per l'utilizzo su piani cartesiani quindi bisogna sostituire questi concetti con degli equivalenti cartesiani; si introduce il concetto di “*corona cartesiana*” basata su quadrati concentrici e linee di taglio. Definendo un analogo cartesiano della finestra W_j come:

$$\tilde{W}_j = \sqrt{\Phi_{j+1}^2 + \Phi_j^2} \quad (\text{A.16})$$

dove Φ è definito come prodotto di finestra passa-basso monodimensionali,

$$\Phi_j(\omega_1, \omega_2) = \phi(2^{-j}\omega_1)\phi(2^{-j}\omega_2) \quad (\text{A.17})$$

Supponendo che la finestra V sia:

$$V_j = V(2^{\lfloor j/2 \rfloor} \frac{\omega_1}{\omega_2}) \quad (\text{A.18})$$

si può definire una finestra “cartesiana” come:

$$\tilde{U}_j = \tilde{W}_j V_j \quad (\text{A.19})$$

\tilde{U}_j isola le frequenze vicino al “cuneo” $\{(\omega_1, \omega_2) : 2^j \leq \omega_1 \leq 2^{j+1}, -2^{-j/2} \leq \frac{\omega_1}{\omega_2} \leq 2^{-j/2}\}$. Introducendo un fascio di rette equispaziate $\tan \theta_l = l \cdot 2^{-\lfloor j/2 \rfloor}$ con $l = -2^{-\lfloor j/2 \rfloor}, \dots, 2^{-\lfloor j/2 \rfloor} - 1$, possiamo definire:

$$\tilde{U}_{j,l} = \tilde{W}_j(\omega) V_j(S_{\theta_l} \omega) \quad (\text{A.20})$$

dove S_{θ} è la matrice di taglio,

$$S_{\theta} = \begin{bmatrix} 1 & 0 \\ -\tan \theta & 1 \end{bmatrix} \quad (\text{A.21})$$

A differenza del caso continuo, gli angoli θ_l non sono equispaziati, ma lo sono le rette [10]. La famiglia $\tilde{U}_{j,l}$ implica una “tegolatura” (*tiling*) concentrica

la cui geometria è mostrata nella Figura A.6.

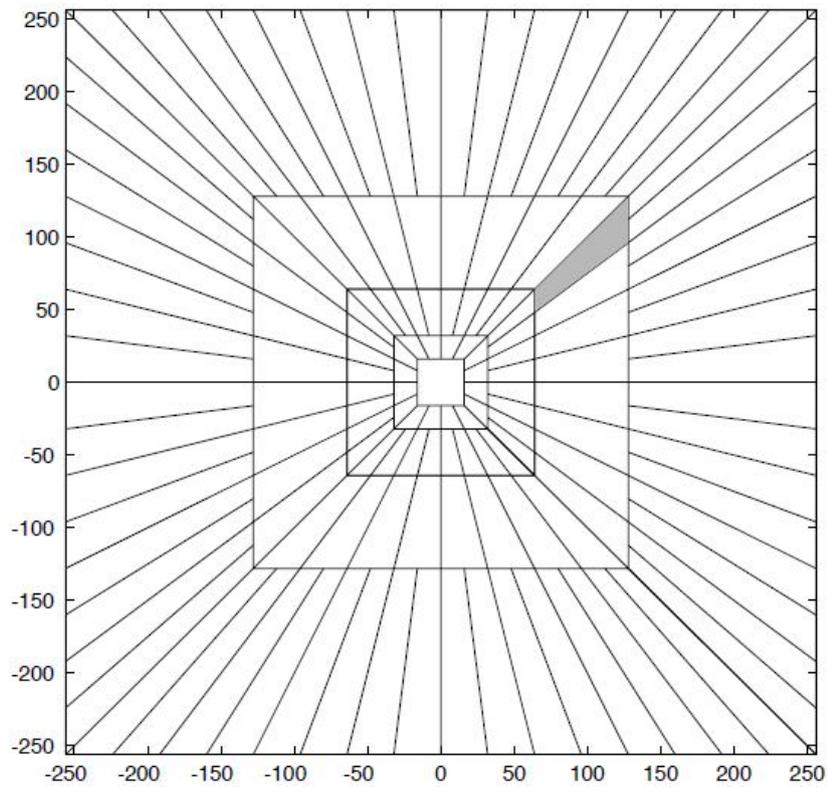


Figura A.6: Suddivisione dello spazio delle frequenze nel dominio digitale. Cortesemente da [10].

Appendice B

Codici MATLAB

B.1 Algoritmo

```
1 function [Image2Dfocus Map_index] = DFF_LocalAnalysis_v3(parameters, map_3D)
2 % AUTHOR: Davide Pollini
3 %
4 %
5 % FILE NAME: DFF_LocalAnalysis_v3.m
6 %
7 % To obtain the final in-focus 2-D image, starting from a stack of
8 % slices where only small areas of each slices are in focus. The
9 % algorithm is based on a local analysis of the first x-y
10 % derivatives of each input image, and a following maximum
11 % analysis of the z-vector of the stack. The final composite image
12 % is built copying for each x-y position the pixel value from the
13 % image point out be the "depth map" built during the maximum
14 % analysis. The algorithm works with both GRAY or RGB images and
15 % the output image is of the same type of the input ones.
16 %
17 % USAGE:
18 % [Image2Dfocus Map_index] = DFF_LocalAnalysis;
19 %
20 % INPUT PARAMETERS:
21 % parameters      All the input parameters are set and explained
22 %                  in the file named: DFF_Parameters.m
23 %
24 % OUTPUT:
25 % FocusIMAGE      Final composite image where all the regions
26 %                  should be in-focus.
27 % DepthMap         2D mask reporting for each x-y pixel position
28 %                  the index of the original image from which is
29 %                  copied the value reported in the output
30 %                  Image2Dfocus.
31 %
32 %
33 if nargin < 1
34     parameters = DFF_Parameters();
35 end
```

Appendice B. Codici MATLAB

B.1. Algoritmo

```
37 if nargin < 2
    map_3D = 'map_3D_off';
39 end

41 flag_rgb2gray_pca = 1;

43 % LOAD IMAGES AND CREATION OF THE STACK
strnum = sprintf('%%.3d',parameters.ImageIndexs(1));
45 name = strcat(parameters.CartellaImmagini,parameters.ImageBaseName,...
    parameters.formato_immagini);
47 OrigImg = imread(name);
if size(OrigImg,3) == 3
49     flag_RGB = 1;
else
51     flag_RGB = 0;
end
53 clear OrigImg name strnum
stop_index = length(parameters.ImageIndexs);
55 if flag_rgb2gray_pca == 1
    if flag_RGB == 1
57         for nf=1:stop_index
            strnum = sprintf('%%.3d',parameters.ImageIndexs(nf));
59             name = strcat(parameters.CartellaImmagini,parameters.ImageBaseName,...
                strnum,parameters.formato_immagini);
61             OrigImg = imread(name);
            R = double(OrigImg(:,:,1));
63             G = double(OrigImg(:,:,2));
            B = double(OrigImg(:,:,3));
65             [NI1, NI2, LR] = princomp(double(R));
            [NI1, NI2, LG] = princomp(double(G));
67             [NI1, NI2, LB] = princomp(double(B));
            L_R(nf) = max(LR);
69             L_G(nf) = max(LG);
            L_B(nf) = max(LB);
71             clear NI1 NI2 LB LG LR B G R OrigImg name strnum
                end
73             L_Rmean = mean(L_R);
            L_Gmean = mean(L_G);
75             L_Bmean = mean(L_B);
            clear L_R L_G L_B
77             FatNorm = L_Rmean + L_Gmean + L_Bmean;
            L_Rw = L_Rmean/FatNorm;
79             L_Gw = L_Gmean/FatNorm;
            L_Bw = L_Bmean/FatNorm;
81             clear FatNorm L_Rmean L_Gmean L_Bmean
                end
83         end

85     Img_RGB{nf} = cell(1,stop_index);
87     for nf=1:stop_index
        strnum = sprintf('%%.3d',parameters.ImageIndexs(nf));
89         name = strcat(parameters.CartellaImmagini,parameters.ImageBaseName,...
            strnum,parameters.formato_immagini);
91         OrigImg = imread(name);
        if flag_RGB == 1
93             if flag_rgb2gray_pca == 1
```

```

    R = double(OrigImg(:,:,1));
95     G = double(OrigImg(:,:,2));
    B = double(OrigImg(:,:,3));
97     Cut_gray(:,:,nf) = L_Rw*R + L_Gw*G + L_Bw*B;
    Img_RGB{nf} = OrigImg;
99     clear R G B
    else
101     Cut_gray(:,:,nf) = double(rgb2gray(OrigImg));
    end
103     else
    Cut_gray(:,:,nf) = double(Orig_img);
105     Img_RGB = [];
    end
107     clear OrigImg name strnum
end
109
% DERIVATIVE X Y
111 for nf=1:stop_index
    Derivative_XY(:,:,nf) = FP_DerivativeXY(Cut_gray(:,:,nf));
113
    % FILTERING APPLIED TO THE DERIVATIVE
115     h = fspecial('average', parameters.SizeFilter);
    Derivative_XY(:,:,nf) = imfilter(Derivative_XY(:,:,nf), h,...
117         'replicate','same', 'conv');

119 end

121 % CREATION OF THE MAP OF MAXIMUM OF DERIVATIVES ALONG THE Z-STACK
[rows, columns, channels] = size(Derivative_XY);
123 Map_index = zeros(rows, columns);
for i = 1:rows
125     for j = 1:columns
        [NI, index] = max(Derivative_XY(i,j,:));
127         Map_index(i,j) = index;
    end
129     clear NI
end
131
% MAJORITY FILTER
133 ROI_size = parameters.MajorityFilter;
pad = (ROI_size-1)/2;
135 Map_index_pad = padarray(Map_index,[pad pad],'replicate','both');
for i = pad+1:rows+pad
137     for j = pad+1:columns+pad
        v = zeros(1,channels);
139         ROI = Map_index_pad(i-pad:i+pad,j-pad:j+pad);
        ROI = ROI(:);
141         for k =1:length(ROI)
            v(ROI(k)) = v(ROI(k))+1;
143         end
        if max(v) >=ceil(ROI_size^2/2)
145             index = find(v==max(v));
            Map_index(i-pad,j-pad) = index;
147         end
        clear v ROI index
149     end
end

```

```
151
152 if isempty(Img_RGB) == 1
153     % CREATION OF THE FUSED IMAGE
154     Focus_gray = zeros(size(Map_index));
155     for i = 1:rows
156         for j = 1:columns
157             Focus_gray(i,j) = Cut_gray(i,j,Map_index(i,j));
158         end
159     end
160     % DISPLAY FUSED IMAGE AND DEPTH IMAGE
161     figure, imshow(Focus_gray,[])
162     figure, imshow(Map_index,[])
163     Image2Dfocus = double(Focus_gray);
164 else
165     % CREATION OF THE RGB FUSED IMAGE
166     Focus_2D_RGB = uint8(zeros(rows, columns, 3));
167     for i = 1:rows
168         for j = 1:columns
169             ind = Map_index(i,j);
170             Focus_2D_RGB(i,j,:) = Img_RGB{ind}(i,j,:);
171         end
172     end
173     % DISPLAY RGB FUSED IMAGE AND DEPTH IMAGE
174     figure, imshow(Focus_2D_RGB,[])
175     figure, imshow(Map_index,[])
176     Image2Dfocus = Focus_2D_RGB;
177 end
178
179 % 3-D VISUALIZATION OF SURFACE
180 if strcmp(map_3D,'map_3D_on') == 1
181     FP_display_surface(Map_index, Image2Dfocus, 1, 10, 0)
182 end
```

B.2 Parametri

```
function parameters = DFF_Parameters()
2 % AUTHOR: Davide Pollini
  %
4 % FILE NAME: DFF_Parameters.m
  %
6 % In this script are reported and explained all the parameters
  % required by the function named: DFF_LocalAnalysis. The
8 % parameters must be set by changing the lines of this script.
  %
10 % USAGE:
  % parameters = DFF_Parameters()
12 %
  % See also DFF_LocalAnalysis,
14
  % Parametri da modificare:
16
  % Nome dell'immagine da caricare
18 parameters.ImageBaseName      = 'Dome1x40_0';

20 % Percorso della cartella contenente le immagini
  parameters.CartellaImmagini    = 'H:\Documenti\università\laurea\
22 \magistrale\Tesi\Immagini\Reali\Mouse\Intestin\';

24 % Formato delle immagini da caricare '.tif', '.jpg', '.bmp'
  parameters.formato_immagini    = '.tif';
26
  % Numero di slice che formano lo stack
28 parameters.ImageIndexs       = [001:1:020];

30 % Dimensione (nxn) del filtro a media applicato sulle derivate
  parameters.SizeFilter          = [15];
32
  % Dimensione ROI del filtro di maggioranza
34 parameters.MajorityFilter      = [5];
```

Bibliografia

- [1] NJ) Adelson Edward H. (Princeton. «Depth-of-focus imaging process method». Brev. 4661986. Apr. 1987.
- [2] F. Aguet, D. Van De Ville e M. Unser. «Model-Based 2.5-D Deconvolution for Extended Depth of Field in Brightfield Microscopy». In: *Image Processing, IEEE Transactions on* 17.7 (lug. 2008), pp. 1144–1153. ISSN: 1057-7149.
- [3] Mark Antunes et al. «All-in-Focus Imaging Using a Series of Images on Different Focal Planes». In: *Image Analysis and Recognition*. A cura di Mohamed Kamel e AurÃ©lio Campilho. Vol. 3656. Springer Berlin / Heidelberg, 2005, pp. 174–181. ISBN: 978-3-540-29069-8.
- [4] A. Ardeshir Goshtasby e Stavri Nikolov. «Guest editorial: Image fusion: Advances in the state of the art». In: *Inf. Fusion* 8 (2 apr. 2007), pp. 114–118. ISSN: 1566-2535.
- [5] A. Bevilacqua, F. Piccinini e A. Gherardi. «Vignetting correction by exploiting an optical microscopy image sequence». In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. Set. 2011, pp. 6166–6169.
- [6] Pascal C. Bradley Andrew P. Bamford. «A One-Pass Extended Depth of Field Algorithm Based on the Over-Complete Discrete Wavelet Transform». In: *Image and Vision Computing '04 New Zealand (IVCNZ'04)*. Akaroa, New Zealand, nov. 2004, pp. 279–284.
- [7] J F Brenner et al. «An automated microscope for cytologic research a preliminary evaluation.» In: *Journal of Histochemistry & Cytochemistry* 24.1 (1976), pp. 100–11.
- [8] Mario A. Bueno et al. «Polychromatic image fusion algorithm and fusion metric for automatized microscopes». In: *Optical Engineering* 44.9, 093201 (2005), p. 093201.

- [9] P.J. Burt e R.J. Kolczynski. «Enhanced image capture through fusion». In: *Computer Vision, 1993. Proceedings., Fourth International Conference on*. Mag. 1993, pp. 173 –182.
- [10] Emmanuel Cande's et al. «Fast Discrete Curvelet Transforms». In: *Multiscale Modeling Simulation* 5.3 (2006), p. 861.
- [11] L. Carozza, A. Bevilacqua e F. Piccinini. «Mosaicing of optical microscope imagery based on visual information». In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. Set. 2011, pp. 6162 –6165.
- [12] L.J. Chipman, T.M. Orr e L.N. Graham. «Wavelets and image fusion». In: *Image Processing, 1995. Proceedings., International Conference on*. Vol. 3. Ott. 1995, 248 –251 vol.3.
- [13] Bernard Desoize e Jean-Claude Jardillier. «Multicellular resistance: a paradigm for clinical resistance?» In: *Critical Reviews in Oncology/Hematology* 36 (2000), pp. 193 –207. ISSN: 1040-8428.
- [14] David L. Donoho e Mark R. Duncan. «Digital Curvelet Transform: Strategy, Implementation and Experiments». In: *in Proc. Aerosense 2000, Wavelet Applications VII*. SPIE, 1999, pp. 12–29.
- [15] Christophe Dubessy et al. «Spheroids in radiobiology and photodynamic therapy». In: *Critical Reviews in Oncology/Hematology* 36 (2000), pp. 179 –192. ISSN: 1040-8428.
- [16] Lawrence Firestone et al. «Comparison of autofocus methods for automated microscopy». In: *Cytometry* 12.3 (1991), pp. 195–206. ISSN: 1097-0320.
- [17] Brigitte Forster et al. «Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images». In: *Microscopy Research and Technique* 65.1-2 (2004), pp. 33–42. ISSN: 1097-0029.
- [18] Gemma e Piella. «A general framework for multiresolution image fusion: from pixels to regions». In: *Information Fusion* 4.4 (2003), pp. 259 –280. ISSN: 1566-2535.
- [19] A. Gherardi, A. Bevilacqua e F. Piccinini. «Illumination field estimation through background detection in optical microscopy». In: *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2011 IEEE Symposium on*. Apr. 2011, pp. 1 –6.

- [20] Noel T. Goldsmith. «Deep Focus; A Digital Image Processing Technique to Produce Improved Focal Depth in Light Microscopy». In: *Image Analysis and Stereology* 19 (2000), pp. 163 –167.
- [21] H. Harms e H. M. Aus. «Comparison of digital focus criteria for a TV microscope system». In: *Cytometry* 5.3 (1984), pp. 236–243. ISSN: 1097-0320.
- [22] M. Hossny, S. Nahavandi e D. Creighton. «A Quadtree Driven Image Fusion Quality Assessment». In: *Industrial Informatics, 2007 5th IEEE International Conference on*. Vol. 1. Giu. 2007, pp. 419 –424.
- [23] Mohammed Hossny, Saeid Nahavandi e Doug Crieghton. «Feature-Based Image Fusion Quality Metrics». In: *Intelligent Robotics and Applications*. A cura di Caihua Xiong et al. Vol. 5314. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, pp. 469–478. ISBN: 978-3-540-88512-2.
- [24] N. Kingsbury. «A dual-tree complex wavelet transform with improved orthogonality and symmetry properties». In: *Image Processing, 2000. Proceedings. 2000 International Conference on*. Vol. 2. Set. 2000, 375 –378 vol.2.
- [25] I. Koren, A. Laine e F. Taylor. «Image fusion using steerable dyadic wavelet transform». In: *Image Processing, 1995. Proceedings., International Conference on*. Vol. 3. Ott. 1995, 232 –235 vol.3.
- [26] Eric Krotkov. «Focusing». In: *International Journal of Computer Vision* 1 (3 1988). 10.1007/BF00127822, pp. 223–237. ISSN: 0920-5691.
- [27] Leoni A. Kunz-Schughart, Marina Kreutz e Ruth Knuechel. «Multicellular spheroids: a three-dimensional in vitro culture system to study tumour biology». In: *International Journal of Experimental Pathology* 79.1 (1998), pp. 1–23. ISSN: 1365-2613.
- [28] John J. Lewis et al. «Pixel- and region-based image fusion with complex wavelets». In: *Information Fusion* 8.2 (2007). <ce:title>Special Issue on Image Fusion: Advances in the State of the Art</ce:title>, pp. 119 –130. ISSN: 1566-2535.
- [29] H. Li, B.S. Manjunath e S.K. Mitra. «Multisensor Image Fusion Using the Wavelet Transform». In: *Graphical Models and Image Processing* 57.3 (1995), pp. 235 –245. ISSN: 1077-3169.

- [30] Shutao Li e Bin Yang. «Multifocus image fusion using region segmentation and spatial frequency». In: *Image and Vision Computing* 26.7 (2008), pp. 971 –979. ISSN: 0262-8856.
- [31] Shutao Li, Bin Yang e Jianwen Hu. «Performance comparison of different multi-resolution transforms for image fusion». In: *Information Fusion* 12.2 (2011), pp. 74 –84. ISSN: 1566-2535.
- [32] Jianwei Ma e G. Plonka. «The Curvelet Transform». In: *Signal Processing Magazine, IEEE* 27.2 (mar. 2010), pp. 118 –133. ISSN: 1053-5888.
- [33] Aamir Saeed Malik e Tae-Sun Choi. «Consideration of illumination effects and optimization of window size for accurate calculation of depth map for 3D shape recovery». In: *Pattern Recognition* 40.1 (2007), pp. 154 –170. ISSN: 0031-3203.
- [34] Wolfgang Mueller-Klieser. «Three-dimensional cell cultures: from molecular mechanisms to clinical applications». In: *American Journal of Physiology - Cell Physiology* 273.4 (1997), pp. C1109–C1123.
- [35] Filippo Nencini et al. «Remote sensing image fusion using the curvelet transform». In: *Information Fusion* 8.2 (2007). <ce:title>Special Issue on Image Fusion: Advances in the State of the Art</ce:title>, pp. 143 –156. ISSN: 1566-2535.
- [36] V.S. Petrovic e C.S. Xydeas. «Gradient-based multiresolution image fusion». In: *Image Processing, IEEE Transactions on* 13.2 (feb. 2004), pp. 228 –237. ISSN: 1057-7149.
- [37] G. Piella e H. Heijmans. «A new quality metric for image fusion». In: *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. Vol. 3. Set. 2003, III –173–6 vol.2.
- [38] R. J. Pieper e A. Korpel. «Image processing for extended depth of field». In: *Appl. Opt.* 22.10 (mag. 1983), pp. 1449–1453.
- [39] Robi Polikar. *The Wavelet Tutorial*. 2004.
- [40] Zhiyun Xue Rick S. Blum e Zhong Zhang. *An Overview of Image Fusion*. A cura di Blum. Taylor e Francis, 2005.
- [41] D. R. Bull S. G. Nikolov P. R. Hill e C. N. Canagarajah. *Wavelets for image fusion*. Kluwer Academic Publishers, 2001.
- [42] I.W. Selesnick, R.G. Baraniuk e N.C. Kingsbury. «The dual-tree complex wavelet transform». In: *Signal Processing Magazine, IEEE* 22.6 (nov. 2005), pp. 123 –151. ISSN: 1053-5888.

- [43] Satoshi A. Sugimoto e Yoshiki Ichioka. «Digital composition of images with increased depth of focus considering depth information». In: *Appl. Opt.* 24.14 (lug. 1985), pp. 2076–2080.
- [44] A. Tesei et al. «Isolation of stem/progenitor cells from normal lung tissue of adult humans». In: *Cell Proliferation* 42.3 (2009), pp. 298–308. ISSN: 1365-2184.
- [45] L. Tessens et al. «Extending the Depth of Field in Microscopy Through Curvelet-Based Frequency-Adaptive Image Fusion». In: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 1. Apr. 2007, pp. I-861 –I-864.
- [46] A.G. Valdecasas et al. «On the extended depth of focus algorithms for bright field microscopy». In: *Micron* 32.6 (2001), pp. 559 –569. ISSN: 0968-4328.
- [47] Qiang Wang, Daren Yu e Yi Shen. «An overview of image fusion metrics». In: *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*. Mag. 2009, pp. 918 –923.
- [48] Zhou Wang e A.C. Bovik. «A universal image quality index». In: *Signal Processing Letters, IEEE* 9.3 (mar. 2002), pp. 81 –84. ISSN: 1070-9908.
- [49] Zhou Wang e A.C. Bovik. «Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures». In: *Signal Processing Magazine, IEEE* 26.1 (gen. 2009), pp. 98 –117. ISSN: 1053-5888.
- [50] Allen M Waxman et al. «Color Night Vision: Opponent Processing in the Fusion of Visible and IR Imagery». In: *Neural Networks* 10.1 (1997), pp. 1 –6. ISSN: 0893-6080.
- [51] Wolfgang e Mueller-Klieser. «Tumor biology and experimental therapeutics». In: *Critical Reviews in Oncology/Hematology* 36 (2000), pp. 123 –139. ISSN: 1040-8428.
- [52] Hui Xie et al. «Image Fusion and 3-D Surface Reconstruction of Microparts Using Complex Valued Wavelet Transforms». In: *Image Processing, 2006 IEEE International Conference on*. Ott. 2006, pp. 2137 –2140.
- [53] Y. Xiong e S.A. Shafer. «Depth from focusing and defocusing». In: *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*. Giu. 1993, pp. 68 –73.

- [54] C.S. Xydeas e V. Petrovic. «Objective image fusion performance measure». In: *Electronics Letters* 36.4 (feb. 2000), pp. 308–309. ISSN: 0013-5194.
- [55] TTE Yeo et al. «Autofocusing for tissue microscopy». In: *Image and Vision Computing* 11.10 (1993), pp. 629–639. ISSN: 0262-8856.
- [56] Zhong Zhang e R S Blum. «A categorization of multiscale-decomposition-based image fusion schemes with a performance study for a digital camera application». In: *Proceedings of the IEEE* 87.8 (1999), pp. 1315–1326.
- [57] Jiang Zhi-guo et al. «A wavelet based algorithm for multi-focus micro-image fusion». In: *Image and Graphics, 2004. Proceedings. Third International Conference on*. Dic. 2004, pp. 176–179.

Ringraziamenti

Voglio spendere parole di ringraziamento per:

- Il Prof. Alessandro Bevilacqua per avermi permesso di collaborare con il suo gruppo di ricerca. Lo avevo conosciuto durante il corso triennale, l'ho rincontrato al corso magistrale e la stima è aumentata di giorno in giorno facendomi appassionare sempre di più alla sua materia e desiderare di poter completare il mio percorso di studio con lui.
- La Dott.ssa Anna Tesei per il contributo e i suggerimenti dati per la stesura del mio lavoro di tesi.
- L'Ing. Filippo Piccinini per avermi accompagnato e sostenuto in tutti questi mesi. Sono contento di averlo conosciuto e di aver collaborato con lui. Siamo entrati subito molto in sintonia tanto da rendere più facile il lavoro. Persona generosa, precisa, disponibile e ricca umanamente tanto da sperare di poter continuare a frequentarci.
- Mia madre che con il suo amore e il suo sostegno mi ha permesso di arrivare sin qui.
- La “zia” Laura per la continua e assidua presenza affettiva, il sostegno morale nonostante la distanza che ci separa; era come averla qui.
- Sandy per avermi sopportato ed essermi stata vicino in questo periodo di studi proteso solo a pensare a me stesso ed al mio traguardo.
- Tutti, ma proprio tutti, gli amici miei e di mia mamma, che non cito per paura di dimenticare qualcuno ma che mi hanno fatto svagare e divertire anche nei momenti più difficili.

Ma se una tesi si può dedicare: questa è per mio padre.

